P139 - Observational Astrophysics
Lab Manual

Lab 1: Deriving the Mass of Jupiter or Saturn

Professor T. Smecker-Hane
Version 4.0
Sept 25, 2007

1 Introduction

The mass of a planet can be determined by measuring the orbital period, τ, the time it takes to complete one orbit, and the orbital radius, A, a constant assuming the orbit is circular, of one of its moons. Doing this independently for a number of moons rather than just one allows you to decrease the error in your derived mass. For Jupiter, there are four easily observable moons. Listed in order from smallest to largest A are: Io, Europa, Ganymede and Callisto. These are commonly referred to as the Galilean moons, because Galileo discovered them in 1610. For Saturn, there are five easily observable moons. Listed in order from smallest to largest A are: Enceladus, Tethys, Dione, Rhea and Titan.

2 Theoretical Framework for the Experiment

Use your knowledge of Newton's laws and classical mechanics to determine the center of mass of the system composed of a planet of mass M and a moon of mass m. Recall that the two bodies both rotate around the center of mass with the period equal to τ. Assume that the moon's orbit is circular, and the distance between the two in the orbital plane is r and $r = A$, where A is a constant for a circular orbit. Now determine τ as a function of A, m and M. Next take the limit of $m << M$, as is the case for all the moons of Jupiter and Saturn, and determine M as a function of A and τ. You will use this to determine the planet's mass once you determine the moon's orbital parameters, A and τ.

Deriving the orbital parameters is complicated by the fact that the moons' orbits are tilted with respect to our line of sight. We define the inclination angle θ of the orbit such that $\theta = 0^\circ$ for an orbital plane that is perpendicular to our line of sight (also known as edge-on) and $\theta = 90^\circ$ for an orbital plane that is face-on.

One of the Euler angles and its transformation matrix, which you should have learned about in your Classical Mechanics class, can be used to transform from the orbital plane to the observed plane. Determine the relationship between the coordinates of the moon in the

orbital plane (x, y), which would be called the "body coordinates" in the language of your Classical Mechanics textbook, and the coordinates on the sky (x', y').

Make the assumption that the motion in the orbital plane is given by

$$\begin{aligned} x(t) &= A\cos\phi \\ y(t) &= A\sin\phi \ , \end{aligned} \tag{1}$$

where the phase ϕ is given by

$$\phi = \frac{2\pi(t - t_0)}{\tau} \ , \tag{2}$$

τ is the orbital period, and t_0 is the first time after $t = 0$ that the maximum distance between the planet and moon occurs. We'll conveniently define $t = 0$ to be the time of our first observation. Then derive the position of the moon in the plane of the sky as a function of time ($x'(t)$, $y'(t)$) and the distance between the planet and moon in the plane of the sky $r'(t)$ assuming that

$$r' = (x'^2 + y'^2)^{1/2} \ , \tag{3}$$

Check your results by plotting ($x'(t)$, $y'(t)$) and $(r'(t), t)$ over a few periods for a few adopted values of A, τ and θ. Check that they agree with the examples shown in Figures 1 and 2.

Question: What feature of the (r', t) plot depends most sensitively on θ? Therefore to get the best observational constraints on θ you'll want to observe during that part of the phase. Based on your first night or two of data, you should estimate when that phase will occur and plan to observe at that time, if it's feasible.

3 Planning Your Observations

The TA will give you an excerpt from the book entitled "Spherical Astronomy" (Green 1998; pages 1–39) that you need to read to familiarize yourself with the two coordinate systems used in astronomy (Equatorial and Horizon) and terms such zenith, hour angle and airmass.

Use one of the available planetarium programs (*TheSky* on the Windows PCs in the MSTB lab, *XEPHEM* on the Linux computer at the Observatory, or *Redshift* on the Astronomy Outreach Windows laptop) to determine the best time of day to observe Saturn or Jupiter. To image Saturn's and Jupiter's moons, which are bright relative to most stars, you can probably begin observing about a half hour after sunset and end a half hour before sunrise. Realize that the time of sunrise and sunset varies over time, and remember that Daylight Savings Time ends (clocks are set back one hour) in late October/early November. Table 1 shows typical times for various months of the year.

The UCI telescope can observe objects with an altitude $\gtrsim 20°$, although the closer the object is to zenith (altitude = $+90°$) the more accurately the telescope will track the motion

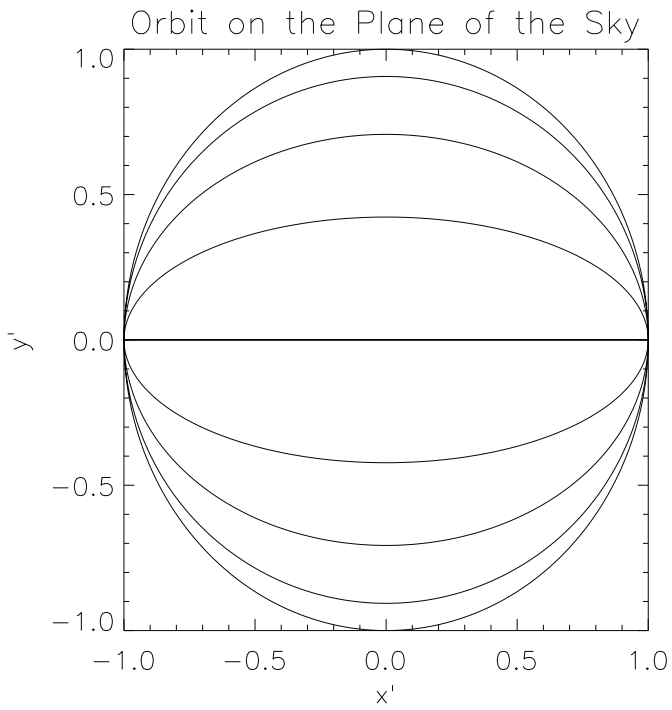

Figure 1: Orbital position (x', y') as a function of the phase ϕ for orbits with five different inclination angles: $\theta = 0, 25, 45, 65$, and 90 degrees.

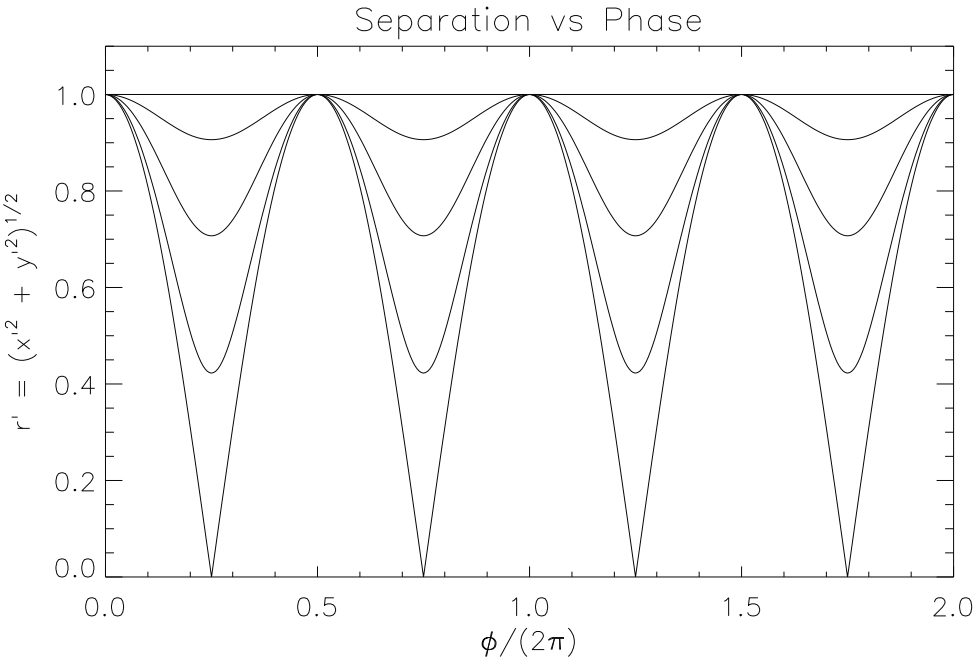

Figure 2: Orbital distance r' as a function of the phase ϕ for orbits with five different inclination angles: $\theta = 0, 25, 45, 65$, and 90 degrees. Two full orbits are plotted.

Table 1: Solar Almanac

Date (dd/mm/yy)	Local Time When ... Sun Sets	Twilight Ends	Twilight Begins	Sun Rises
01/01/2007	16:57	18:23	5:27	6:53
01/02/2007	17:25	18:48	5:22	6:45
01/03/2007	17:50	19:11	4:56	6:16
01/04/2007	19:14	20:36	5:13	6:35
01/05/2007	19:37	21:06	4:30	5:59
01/10/2007	18:39	19:59	5:24	6:44
01/11/2007	17:02	18:24	4:47	6:08
01/12/2007	16:46	18:11	5:10	6:36

of the object across the sky. You'll want a window at least one hour in length to observe the planet, but something more like three hours is best. Therefore use the program to find the best or at least acceptable time of night for you to perform your observations.

Tables 2 and 3 show some examples to guide you. To observe Saturn in Fall quarter, you should wait until late in the quarter, and you'll have to get up early in the morning to catch Saturn after it rises, but before the Sun rises. Notice that as the years go by, students will find it more difficult to observe Saturn in Fall quarter, because it will rise later in the night. When that happens we can supply you with previous year's data to do the project.

To observe Jupiter in Fall quarter, you should plan to observe early in the quarter and you must begin observing as soon as the sky gets dark. You'll be limited in the amount of time you have to observe by the fact that Jupiter will set before the end of the night. As opposed to Saturn, as the years go by, students will have more time to observe Jupiter on Fall nights, because it sets later in the night each year.

Also realize that you can't observe the faint moons of a planet if it is too close to the Earth's Moon, because the Moon's brightness will swamp its light. So check the Moon's location relative to the planet and its illumination fraction. If it is within 10° of the planet you may want to wait a day or so for the Moon to change its location in the sky.

4 Obtaining the Data

The data for this project will be obtained with the UCI Observatory's 24" telescope and the ST-9E CCD camera permanently installed inside the dome of the campus observatory. (In some cases, you may be asked to use the portable 8" Meade LX200/GPS telescope and the ST-7XE CCD.) The 24" telescope and camera are controlled by a Linux computer, so you should get familiar with Linux. For example, read the "LINUX for Dummies" or equivalent book. To prepare for making the observations, you must read the following:

Table 2: Saturnian Altitudes

Date (dd/mm/yy)	**Local Time**	**Altitude of Saturn** (°)
01/02/07	20:00	20 (in the East and Rising)
01/02/07	21:00	30
01/02/07	22:00	42
01/02/07	23:00	54
01/11/07	3:00	15 (in the East and Rising)
01/11/07	4:00	27
01/11/07	5:00	39
01/11/07	6:00	51
01/12/07	3:00	37 (in the East and Rising)
01/12/07	4:00	49
01/12/07	5:00	59
01/12/07	6:00	65
01/12/08	3:00	26 (in the East and Rising)
01/12/08	4:00	38
01/12/08	5:00	49
01/12/08	6:00	57

Table 3: Jovian Altitudes

Date (dd/mm/yy)	**Local Time**	**Altitude of Jupiter** (°)
24/09/07	19:00	27 (in the Southwest and Setting)
24/09/07	20:00	19
24/09/08	19:00	33
24/09/08	20:00	31
24/09/08	21:00	26
15/10/08	19:00	30
15/10/08	19:00	24
01/04/07	2:00	20 (in East and rising)
01/04/07	3:00	27
01/04/07	4:00	32

Readings:

1. D.S. Birney, G. Gonzalea, D. Oesper (2006), *Observational Astronomy, Second Edition*, Cambridge University Press, pgs. 159–181
2. Massey & Jacoby (1992), "CCD Data: The Good, the Bad, and the Ugly", in *Astronomical CCD Observing and Reduction Techniques*, ed. S. B. Howell (ASP: San Francisco), Astron. Soc. of the Pacific Conf. Series, 23, 240–253
3. UCI Observatory Manual
4. CCDAuto Manual
5. Imaging Checklist

Photocopies of items 1 and 2 will be given to you by your TA. They will introduce you to CCD detectors and the image processing steps you'll need to perform on the images. Items 3 through 5 are available on the UCI Observatory web site, `http://www.physics.uci.edu/~observat`, under "Technical Information" for the 24" telescope. Item 3 is the manual describing the telescope and the software program called ucirob that is used to control it, item 4 is the manual describing CCDAuto, which is the program used to operate the CCD camera, and item 5 is a step-by-step checklist that you'll use to operate the telescope and obtain images. You should read the manuals thoroughly and be *very* familiar with them before you attempt to go observing with the Imaging Checklist in hand.

You will go out to the observatory with your TA on one night in order for him/her to familiarize you with the telescope and CCD camera. If you have a favorite Messier object[1] that's up in the sky that night you can practice by taking images of it. The second time you go to the observatory will be to take your first set of images. Your TA will be there with you throughout the night. On subsequent nights, your TA will be there at the start of the night, but he/she will leave after you are setup and are successfully taking images. On those nights, you'll have to close the observatory yourself. Make sure you have the TA's cell phone number in case anything goes wrong. In an emergency, you can also phone Professor Tammy Smecker-Hane, but you should try your TA first.

TA Name: _____

TA Email: _____

TA Home/Cell Phone #: _____

Home Phone # for Prof. Smecker-Hane: _____

[1]The Messier Catalog was compiled by the French astronomer Charles Messier in the late 1700s, and it consists of 106 objects that were resolved with his telescope (extended rather than point sources) and did not change position with respect to the stars over time. He was searching for comets, and they are resolved objects that don't move with respect to the stars. The objects in Messier's Catalog turned out to be relatively nearby star clusters, nebulae and galaxies. For images and details about these objects, see `http://www.seds.org/messier`.

For this project, you will take a series of images on at least 4 to 5 nights. Since the period of Saturn's largest moon, Titan, is $\sim$ 15 days and the period of Jupiter's Callisto is $\sim$ 17 days, your observations should cover the span of $\approx$ 2 weeks if you want to map the entire orbits of those satellites.

When you first arrive at the telescope, log onto the computer (the user name and password will be supplied by the TA) and create a subdirectory for your lab group (e.g., ''saturn06") under the advlab directory. Within this directory, you will create a sub-directory for each night you observe (e.g., "02-20-2007"). In these sub-directories, create three additional directories to store your images: "darks", "flats", and "images". If you start the telescope software, called ucirob, from that night's directory, the software will automatically store each image type in its proper sub-directory.

ucirob username: _____

ucirob password: _____

4.1 Science Images

When observing the motion of the moons within one night, your series of exposures should span 2 to 4 hours. The more, the better. This will allow you to observe Saturn's moons that are temporarily transiting the rings, and it will help you determine which moon is which, because the inner moons have shorter periods. Also, you need to constrain the motions over a few hours baseline on at least one night if you want to avoid problems with aliasing, which is the fact that very sparsely sampled data can be consistent with a solution of a period τ and integer fractions of that period, e.g., $\tau/2$, $\tau/3$, ... τ/n.

Because the moons don't move very rapidly, you can space your images out and take a set of images, for example, once every 15 minutes. Due to the extreme difference in brightness between the planets and their moons, you will not be able to accurately measure the positions of both the planet and the moons in the same image – when taking a long enough exposure to see the moons, the planet will be saturated. For this reason, you will need to take 2 exposures each time you image the moons. The first exposure should be $\gtrsim 1$ second in length in order to well expose all the moons. The second should be ≈ 0.1 seconds, which is the minimum exposure time for the ST-9E CCD, in order to not saturate the planet and to accurately measure its position. See Figures 3 and 4 for an example of a 0.1 sec image displayed at different levels. These exposure times are long enough that Titan, the largest and brightest moon, will be visible in both images and can be used as a reference point to determine any shift in x and y between the exposures. These images, referred to as "science images" should be stored in the "images" directory.

4.2 Calibration Images

In addition to the images you take of the planet and its moons, you will need to take a few images for calibration purposes. The first set of images that you will take are called "dark"

Figure 3: A 0.1 sec R-band image of Saturn displayed at the default level using a linear scaling of the pixel values.

Figure 4: This is the same image shown in Figure 3, except it was displayed after logarithmically scaling the pixel values and displaying with a specific z1 and z2, which is designed to enhance the brightness of the faint features and allow you to see the four moons of Saturn in the image.

images. A CCD operates via the photoelectric effect, where electrons in a silicon chip are freed from the atom when struck by an incoming photon, creating an electric current. An off-chip circuit counts these electrons, thereby counting the number of photons. However, not all the electrons that are freed from the atom are due to incoming photons. Thermal energy can also free some electrons. The number of electrons freed by thermal energy is minimized by cooling the CCD, but cannot be completely eliminated. In order to count these, an image is taken with the camera shutter closed. This is the dark image. To correct for this effect, the dark image is subtracted from each science image. The exposure time of the dark image should match the exposure time of the science image. If the dark images are properly stored in the correct "darks" directory, the telescope software will automatically find them and perform the dark correction before saving the science image.

In the CCDAuto software, which is used to take the images, you can specify using the saved dark images rather than taking a new one each time the exposure time is changed. This will save time during the night, but beware that the dark count will vary with the temperature of the CCD, so you must be sure to take the darks with the CCD set to the same temperature that you'll use when taking the science data.

The other type of calibration image you will take is known as a "flat field" image. The number of valence electrons an atom releases when hit by an incoming photon varies from pixel to pixel. In order to correct for this, an image is taken of an evenly illuminated surface, such as the out-of-focus inside of the dome or the twilight sky. In order to make the flat as evenly illuminated as possible, a series of these images are taken, and then combined into a single image (usually by taking the median of the individual images with rejection of outlying values) and stored in the "flats" directory. The procedure performed to use this image is described in Section 5.2.

4.3 Transporting Your Data

After taking your images on the observatory computer, which is ucirob.ps.uci.edu, you'll want to transport them to the computer on which you plan to analyze your data. Your TA will set you up with an account on the Linux computers in either the MSTB Room 110 lab or in the Astrophysics Computer Room in FRH Room 4109G. While you're at the telescope, pack up your data into a single tar file:[2]

```
ucirob> tar cvf ../night1_data.tar *
```

Then copy it over to, for example, your account on computer abell.ps.uci.edu

```
ucirob> scp ../night1_data.tar NAME@abell.ps.uci.edu:~/night1_data.tar
```

where `NAME` is your user name and "~" represents your home directory. You'll be asked to give the password for your account on abell.ps.uci.edu. Then be sure to delete the extra copy of your data, because disk space on the observatory computer is limited.

```
ucirob> rm ../night1_data.tar
```

[2]The "`ucirob>` " shown when giving examples of Linux commands is merely the prompt that appears on the command line and is not to be typed.

Once you log onto abell.ps.uci.edu, you can unpack the tar file with

```
abell> tar xvf night1_data.tar
```

Its a good idea to save the tar file as your backup copy of the raw data.

5 Data Reduction and Analysis

This section describes the data reduction process and how to accurately measure the position of objects on an image.

5.1 Processing the CCD Images

You will first need to process the CCD images before analyzing them. There are two steps in the process. The first step is subtracting the average dark image. This step you will do automatically in CCDAuto as the images are being taken (see Section 4.2). The next step is to flat field your images. This is described in Section 5.2. Once your images are prepared for analysis, you will measure the positions of the planet and the moons on each, described in Section 5.4. You will do this using the Image Reduction and Analysis Facility (**IRAF**)[3], which is a software package operated on the Linux computers in the MSTB Room 110 computer lab or in the Astrophysics Computer Lab in FRH Room 4109G. Listed throughout this section are examples of the IRAF commands you will use to analyze your data[4].

5.2 Flat Fielding

On the Linux computer, open the image display software by typing "ds9" or "ximtool". Enter your IRAF directory (you *must* be in the IRAF directory in order to start IRAF so the correct login files can be read; the TA will help you set up your account) and start IRAF by typing "cl". Now to display an image, do

```
cl> display image001.fits 1
```

The software automatically determines the scaling and contrast (meaning what pixel values, called z1 and z2, in the image is colored blackest black and whitest white, respectively; note that all values $\leq$ z1 will appear black and all values $\geq$ z2 will appear white). If you want to change this simply do the following,

```
cl> display image001.fits 1 zr- zs- z1=Z1 z2=Z2
```

where `Z1` and `Z2` are the numerical values that you want to use.

[3]**IRAF** is freely distributed by the National Optical Astronomy Observatories, which is operated by the Association of Universities for Research in Astronomy, Inc., under cooperative agreement with the National Science Foundation.

[4]The "`cl>` " shown when giving examples of IRAF commands is merely the prompt that IRAF uses and is not to be typed.

When you display the image, the `1` specifies the frame the image is to be displayed in. For example, ximtool and ds9 will allow you to load 4 images at one time, each in a different frame, and you can blink back and forth between the images.

When you first display a raw science image, notice how there are patterns in the image that do not correspond to anything in the sky. These are due to the detector response described in Section 4.2, or they are due to dust on the CCD window or on the filter that are out of focus and thus donut-shaped. To correct for these effects, you simply divide your science image by your flat field. This is done using IRAF's IMARITH routine, which performs *arith*metic on an *im*age or series of *im*ages:

```
cl> imarith image001.fits / flat001.fits proc001.fits
```

The syntax is simple. You invoke the IMARITH command, then type the rest like a simple equation, without the "=" sign. In this case, we are dividing image001.fits by flat001.fits, and storing the result as proc001.fits.

5.3 Inventorying Your Data

The first step in analyzing your images is to inventory your data and create a file with the names (*.fits), date of observation (DATE-OBS), universal time of observation (UT), Heliocentric Julian Date (HJD), and exposure time (EXPOSURE) of each image. Note that the exposure time has units of seconds and the HJD is in days. All those items are listed in the header of each image. To read the header in IRAF, do

```
cl> imhead NAME.fits l+ | page
```

where `NAME` is the image name. You'll be able to page through the image header by hitting "enter" to go line by line or hitting the space bar to go to the next page of information, if there's more than fits on a single screen. To get help on any IRAF task do

```
cl> help TASK
```

where `TASK` is the name of the task.

Now get the information from the headers and send it to a file. In IRAF, do

```
cl> hedit *.fits date-obs,ut,hjd,exposure .  > file1
```

The HEDIT command is usually used to *edit* information in the *h*eader of an image. The "*" is a Linux "wildcard", meaning we are listing information for any file ending in ".fits". The next set of parameters are the field names for the information we want to list, followed by the "." parameter, which tells the HEDIT command that we are only listing the information, not editing it. The ">" tells IRAF we wish to pipe this information to a file, in this case, one called `file1`. To see what's in `file1` type

```
cl> page file1
```

We have an IDL[5] routine that will take this file and create a column format output in which each line contains the image name, and the rest of the items of information. To use it in

[5]Interactive Data Language (IDL) is a proprietary software suite with powerful analysis and graphical capabilities. Along with IRAF, its commonly used by astrophysicists. You can purchase a student version,

Table 4: Common IMEXAM Keystrokes

Keystroke	Action
r	Creates a plot of intensity vs. radius, fitting it with either a Gaussian or a Moffat function
e	Creates a contour plot, centered around the coordinates of the mouse at the time of the keystroke
m	Gives statistics of the pixels in a box centered on the cursor with the dimensions of the box being given by the `ncstat` and `nlstat` parameters
q	Quits the IMEXAM routine

IDL, do

```
IDL> gather_hdr_items, 'file1', 4, 'images'
```

and check the output in the file called images. You can use this as a basis for building a file that will contain all your observational data. You might want to compose a master file with all the observations and positions of each moon (but be aware that not all moons will be observable on each image) or a file for each moon with its data.

You can look at the IDL routine gather_hdr_items.pro shown in Appendix B to familiarize yourself with the syntax of IDL. It's very much like Fortran or C, but its power is that it has many excellent analysis routines and graphical capabilities. Appendix D shows you how to generate a postscript plot for inclusion in your lab report. You can use the convert command in Linux if you need to convert it to another format, such as jpg.

5.4 Measuring Positions

Here we explain how to accurately measure the (x, y) position of an object on an image, assuming that it is adequately well described by a two dimensional Gaussian or Moffat function. First load a flat-fielded image into the image display (again using the DISPLAY command). You can then use IRAF's IMEXAM routine to *exam*ine the *im*age:

```
cl> imexam
```

This routine is utilized by moving the cursor over the displayed image and typing a keystroke to perform certain calculations. Listed in Table 4 are several keystrokes you might find useful. You will mainly be utilizing the "r" keystroke, which will fit an image enhancement (such as a moon or a star) with a Gaussian or a Moffat function. The coordinate given as the center of this function is the center of the enhancement.

but its very limited in scope in that it cannot write output files. We suggest using our departmental license on the Linux computers in the MSTB Room 110 lab or in the Astrophysics Computer Room in FRH Room 4109G. In each room there is a brief manual entitled *IDL Basics* that will introduce you to IDL. If you don't see it, ask your TA for a copy. Most students find IDL very easy to use.

In order to get the best calculation for the center of the moon or planet, you will need to *e*dit some of the *param*eters for the "r" keystroke in IMEXAM. To do this, type the command EPARAM followed the by the name of the routine. For example, to edit the parameters of the "r" keystroke in IMEXAM, type:

```
cl> eparam rimexam
```

Doing this takes you to the list of parameters of the specified routine, which you can navigate and edit. Typing a "?" in the field for a parameter brings up a short help menu for that parameter. To save the changes you make, type ":w". To exit, type ":q".

Shown below is a list of parameters for the "r" keystroke of the IMEXAM routine, which you will use for the bulk of your analysis. The first four parameters deal with the labels of the resulting plot, and are unimportant for this project, as are many of the other parameters. The parameters most relevant to this project are summarized on the following page.

```
                Image Reduction and Analysis Facility
 PACKAGE  =  tv
    TASK  =  rimexam

(banner   =          yes)  Standard banner
(title    =             )  Title
(xlabel   =       Radius)  X-axis label
(ylabel   =  Pixel Value)  Y-axis label
(fitplot  =          yes)  Overplot profile fit?
(fittype  =       moffat)  Profile type to fit
(center   =          yes)  Center object in aperture?
(backgro  =          yes)  Fit and subtract background?
(radius   =          17.)  Object radius
(buffer   =           5.)  Background buffer width
(width    =          17.)  Background width
(iterati  =            5)  Number of radius adjustment iterations
(xorder   =            0)  Background x order
(yorder   =            0)  Background y order
(magzero  =          25.)  Magnitude zero point
(beta     =        INDEF)  Moffat beta parameter
(rplot    =          30.)  Plotting radius
(x1       =        INDEF)  X-axis window limit
(x2       =        INDEF)  X-axis window limit
(y1       =        INDEF)  Y-axis window limit
(y2       =        INDEF)  Y-axis window limit
(pointmo  =          yes)  plot points instead of lines?
(marker   =         plus)  point marker character?
(szmarke  =           1.)  marker size
(logx     =           no)  log scale x-axis
(logy     =           no)  log scale y-axis
(box      =          yes)  draw box around periphery of window
(ticklab  =          yes)  label tick marks
(majrx    =            5)  number of major divisions along x grid
(minrx    =            5)  number of minor divisions along x grid
(majry    =            5)  number of major divisions along y grid
(minry    =            5)  number of minor divisions along y grid
(round    =           no)  round axes to nice values?
(mode     =           ql)
```

5.4.1 FITPLOT, FITTYPE, and BETA

As mentioned in Table 4, the "r" keystroke creates a plot of intensity versus radius and fits that plot with either a Gaussian or a Moffat function. Before plotting, the sky background in a circular annulus with inner and outer radii given by (`radius` + `buffer`) and (`radius` + `buffer` + `width`), respectively, is calculated and subtracted from the pixel values. Either a Gaussian or Moffat is then fit to the remaining intensity as function of radius. Which function you adopt will depend on which one fits your image quality better. For example, elongated images, due to tracking problems, might be better fit with the Moffat function. The FITPLOT parameter toggles whether or not to overplot the fit, while the FITTYPE parameter selects which function to use such as the Gaussian or Moffat functions.

The Gaussian function is given by

$$G(x,y) = I_0 \exp(-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}) \ , \tag{4}$$

where I_0 is the central intensity (after background subtraction) and σ is a free parameter determined by the best fit. The routine then quotes the full-width-at-half-maximum (FWHM) of the profile, which is given by FWHM $= 2\sqrt{2\ln 2}\sigma \approx 2.355\sigma$.

The Moffat function is given by

$$M(x,y) = \frac{I_0}{\left(1 + \frac{(x-x_0)^2}{\alpha_x^2} + \frac{(y-y_0)^2}{\alpha_y^2} + \alpha_{xy}(x-x_0)(y-y_0)\right)^{\beta}} \ , \tag{5}$$

where I_0, the α's, and β are free parameters determined by the best fit. Both the Gaussian and Moffat functions also return the calculated center of the profile x_0, y_0 (see Section 5.4.2), which you will adopt as the position of the moon or planet.

5.4.2 CENTER, RADIUS, and RPLOT

The RADIUS parameter sets the radius to which the fit is performed, although the Gaussian or Moffat function that is fit to the profile may be overplotted to a higher radius by setting the RPLOT parameter. The center of the enhancement you are measuring is initially defined by the cursor position. Setting the CENTER parameter allows the center to be adjusted when performing the fit, so it is crucial to set CENTER to "yes". You should set the RADIUS parameter high enough that the moon or planet fits within it. However, setting it too high will cause too much of the fit to be determined by the background, and could cause a nearby star to contaminate the fit. Either one of these would cause an error in the calculation of the moon's position.

5.4.3 BACKGROUND, BUFFER, and WIDTH

These three parameters control how much area on the sky IRAF uses to fit and subtract background light when performing the fit described in Section 5.4.1. The BACKGROUND parameter toggles whether or not IRAF fits and subtracts the background, which you want to do, so set this parameter to "yes". The background is calculated from a circular annulus around the object that's being fit, in this case, Jupiter or Saturn or one of their moons. The inner radius of this annulus is the sum of the RADIUS parameter (see Section 5.4.2) and BUFFER. The width of the annulus is set by the WIDTH parameter.

5.4.4 XORDER and YORDER

These two parameters determine *how* IRAF subtracts the sky background. Setting either or both to zero causes IRAF to calculate a median value in the annulus described in Section 5.4.3 and subtract that value. If the moon you are attempting to fit is in an isolated area on your image, this would be good enough. However, if the moon is near enough to the planet, for example, this may not be the best method, since the sky background would not be roughly constant across your fitting aperture or sky annulus. In this case, it may be better to set these parameters to something greater than zero, to account for this change in the sky background. In this case, the value given to these parameters is the number of terms in a polynomial that is fit to the background in the x and y directions.

5.5 Data Tables

In your lab report, you should include a log of your observations listing date, time, weather conditions, moon phase, etc. In addition, for each moon, provide a data table for the longer exposures listing the exposure name, HJD and the distance from the planet to the moon (in pixels).

6 Analysis of the Orbital Parameters

Once you have compiled a file that contains a list of the relevant data for a specific moon, which at a minimum includes the Heliocentric Julian Date (HJD) and the distance in pixels between the moon and the planet (r'), you will want to use a computer software routine to find the best fitting orbital solution to derive A, τ, ϕ_0 and θ (see Section 2 for their definitions). We recommend using IDL, because it's the software that your TA will know how to use and will be able to support you working in it. However you are free to use alternative software, but realize that your TA will likely not know how to use it so you'll be own your own.

Caution: The Heliocentric Julian Date, HJD, must be read as a double precision floating point number because it has 16 digits!

To find the best fitting orbital parameters, we suggest using the CURVEFIT routine in IDL. We show the help page for it in Appendix E. To get a help page in IDL for a standard IDL routine, type "?" followed by the command name:

```
IDL> ?  curvefit
```

Typing just the "?" brings up IDL's general help page. CURVEFIT uses a gradient-expansion algorithm to compute a non-linear least squares fit to a user-supplied function with an arbitrary number of parameters. Your user-supplied function will return the value of r' and the partial derivatives of r' with respect to the variables A, τ, ϕ and θ, given t and a vector with the variables A, τ, ϕ and θ. Using the equations you developed in Section 2, compute the partial derivatives of r'. Note that you can use CURVEFIT without supplying the partial derivatives, but in this case since they are analytic functions you might as well supply them and converge on the correct solution faster. An initial estimate of the solution must be given at the start so you'll have to get a rough idea of the solution by plotting r' verses t. Iterations are performed until the chi squared changes by less than a specified amount or until the maximum number of iterations is reached.

To learn to use CURVEFIT, you should try out the example that's given in Appendix E and make sure you understand how it works. Once you've done this, ask your TA to use the IDL routine "generate_orbit_data.pro" to generate some fake data for you to analyze. This way you can be working on your analysis routines while you're collecting data. Once you have a solution for the fake data set, check the answer with the TA.

CURVEFIT returns the best fitting solution: A, τ, ϕ and θ, the estimated error in them, and the chi square statistic. A good lab report should include a thorough discussion of these. You should provide a data table listing these results for each moon. In addition, your lab report should include plots of your data and the best fit orbital solution. Needless to say, all plots should contain an appropriate title and labels on both axes, and be fully described in the text. You should also show a sample calculation of the mass of Jupiter or Saturn for one of the moons.

Note that in order to convert the distances in pixels to arcseconds on the sky, you need to know the pixel scale of the CCD. Look it up on the UCI Observatory web page under "Technical Information" / "Instrument Information" / "CCD Characteristics". In addition, in order to convert distances in arcseconds to meters, you will need to know the distance from Earth to Jupiter or Saturn on the dates of your observations. This distance varies throughout the year and your TA will email you the distance once you send him/her your observation dates.

Distance to planet: _____

Once you've done the orbital analysis for each moon, calculate the mass of the planet (for comparison purposes, give the answer in units of kg and in Earth masses) and the derived error. Then analyze the results in a comprehensive manner. Did you get similar inclination angles for the moons? Combine your independent measurement of the planet's mass using the N moons (use weights that reflects the derived errors) to derive your final estimate of the planet's mass. Once you are done, your TA will give you the appendices from Fix (2007) that list the orbital parameters of the moons and the mass of the planet so you can compute

your discrepancies. Discuss your sources of error.

Extra Credit: By doing surface photometry of the rings of Saturn, derive their inclination angle. How does it compare with the inclination angles that you derive from the moons? Should they be similar? Why or why not?

7 References

D.S. Birney, G. Gonzalea, D. Oesper (2006), *Observational Astronomy, Second Edition*, Cambridge University Press, 159–181

Green, R. M. 1988, *Spherical Astronomy*, (Cambridge Univ. Press; Cambridge), 1–39

Massey, P. & Jacoby, G. H. 1992, "CCD Data: The Good, The Bad, and the Ugly", in *Astronomical CCD Observing and Reduction Techniques*, ASP Conf. Series v. 23, 240–257

Fix, J. D. 2007, "Appendix 7 - Satellites of Jupiter & Satellites of Saturn", in *Astronomy: Journey to the Cosmic Frontier (Third Edition)*, (New York: Mc Graw-Hill), A–4 – A–5

A Notes on using IDL

Unlike IRAF, IDL can be started in any directory. To do this, simply type 'idl' at the command prompt. It is easiest to chage to the directory where most of your files are located before starting IDL.

```
abell> cd ~/data
abell> idl
```

To compile a routine, give the '.run' command followed by the name of the routine. If you give all your routines the default IDL extenstion ".pro", you do not need to type it.

```
IDL> .run gather_hdr_items
IDL> gather_hdr_items,'images', 4, 'images.dat'
```

If you are compiling a routine that is not in the directory you started IDL in, you will have give the entire pathname when compiling it. The same is true if you are accessing a file that is in a different directory. However, once a routine is compiled, you do not need to specify the directory name when you run it.

```
abell> cd ~/moredata
abell> idl
IDL> .run ../data/gather_hdr_items
IDL> gather_hdr_items,'../data/images', 4, '../data/images.dat'
```

You can see where it is advantageous to run IDL from the proper directory. The '..' and ' ~ ' used above are used to navigate the file system. Typing '..' changes to the directory above the current one, while ' ~ ' changes to the home directory. For the examples above, the routines and 'images' files were stored in:

```
/home/student1/data
```

while IDL was started in that directory in the first example, and started in:

```
/home/student1/moredata
```

in the second example.

If your program in IDL should crash, you will see an error message similar to the one below and you will first need to return to the "main" program before recompiling the routine. This is done with the 'retall' command:

```
% PRINT: Variable is undefined:  NOVARIABLE.
% Execution halted at:  GATHER_HEADER_ITEMS 492 gather_header_items.pro
%                       $MAIN$
IDL> retall
```

At this point, you should debug your program, and recompile.

To pull up the on-line help interface for IDL, type '?' at the IDL command line.

B IDL Syntax

This shows the IDL routine called gather_hdr_items.pro, which illustrates the syntax used in IDL programming.

```
pro gather_hdr_items, finput, nitems, foutput

; This IDL routine assumes that finput is a file that was
; produced by IRAF's hedit with nitems listed for each image
; and it outputs foutput which contains nitems+1 columns that
; list the image name (assume the .fits extension) and the
; nitems from the header. The format for printing each image's
; items is given in format and should include the image name
; (character string) and whatever format is appropriate for
; the nitems.
;
; IDL> gather_hdr_items, 'images', 4, 'images.dat'
;
;-------------------------------------------------------------

 openr,f1,finput,/get_lun
 openw,f2,foutput,/get_lun
 s = ''
 var = strarr(nitems)

 while not eof(f1) do begin
    i = 0
    readf,f1,s
    p = strpos(s,'.fits')
    im_name = strtrim(strmid(s,0,p), 2)
    p = strpos(s,'=')
    var(i) = strtrim(strmid(s,p+1,999),2)
    for i = 1, nitems-1 do begin
       readf,f1,s
       p = strpos(s,'=')
       var(i) = strtrim(strmid(s,p+1,999),2)
    endfor
    printf,f2,im_name,var
 endwhile

 free_lun,f1
 free_lun,f2

return
end
```

C Figures in IDL

This appendix shows a sample IDL routine that reads in data from a file in a specific format, then plots that data and labels the plot. The sample plot is also shown.

```
pro plot_sample

; This IDL routine reads in sample data from a file assuming the
; first column in the file is x and the second column is f(x).
; The data is then plotted to the screen.  The function f in
; this case is a Gaussian.  The centroid and fwhm of the Gaussian
; are calculated by the routine and labeled on the plot.
;
; IDL> plot_sample
;
;-------------------------------------------------------------

 ; Set the character sizes for our plot
 !p.charsize  = 2.
 !p.charthick = 2.
 !p.thick     = 2.

 ; The read_columns routine reads in data from a file assuming
 ; the file has the data organized in columns.  In this case,
 ; there are 2 columns in the given file, and 0 lines are to be
 ; ignored at the beginning of the file.  A sample line is:
 ;      0.920000      1.14957
 ; The first column is a floating point number 13 characters long,
 ; 6 of which come after the decimal point; the second column is
 ; also 13 characters long, but only 5 come after the decimal.
 ; IDL requires the format definition to be inside '()'.  The
 ; '$' indicates that the line of code is continued on the next
 ; line of text.
 data = read_columns(2,'~/alab_Saturn/sample_data/gauss.dat',0, $
                     '(f13.6,f13.5)')

 ; Set x to the first column and y to the second
 x = data(0,*)
 y = data(1,*)

 ; Find the minimum and maximum values
 xmin = min(x)
 ymin = min(y)
 xmax = max(x)
 ymax = max(y)
```

```
 ; Plot the data, specifying black (color #8) and a certain range
 ; in x and y.  Not specifying a line style or plotting symbol
 ; will cause IDL to use its default parameters, which connect
 ; successive points with a solid line.
 plot,x,y,XRANGE=[xmin,xmax],YRANGE=[ymin-1.,ymax+2.],COLOR=8., $
      XTITLE='x',YTITLE='f(x)',TITLE='Sample Figure'

 ; Find where y is at a maximum and overplot a dashed line (plotting
 ; line style #2) at the corresponding x value.  The syntax is the
 ; same as for plot, except some keywords and not allowed.
 ymax_loc = where(y eq ymax)
 oplot,[x(ymax_loc),x(ymax_loc)],[ymin-1.,ymax+2.],COLOR=8.,LINESTYLE=2.

 ; Find where y is equal to half its maximum - this is tricky since
 ; the data is discrete...
 yhalf        = ymax / 2.
 yhigh_pts    = where(y ge yhalf)
 yhalf_loc    = fltarr(2)

 ; ...need to find where y first goes above its half max value,
 ; and then where it drops back below for good
 npts_above   = n_elements(yhigh_pts)
 yhalf_loc(0) = yhigh_pts(0)
 yhalf_loc(1) = yhigh_pts(npts_above - 1.)       ; Zero-based array
 xhalf        = x(yhalf_loc)

 ; Overplot the fwhm as a solid line (plotting line style #0, which
 ; is the default)
 oplot,[xhalf(0),xhalf(1)],[yhalf,yhalf],COLOR=8.

 ; Use the XYOUTS procedure to label the plot.
 message1 = 'Peak = (' + string(x(ymax_loc),format='(f3.1)') + ', ' + $
                         string(ymax,       format='(f4.1)') + ')'
 xyouts,x(ymax_loc)+0.25,ymax+1.,message1,COLOR=8.

 message2 = 'FWHM = ' + string(xhalf(1) - xhalf(0),format='(f4.2)')
 xyouts,xhalf(1)+0.25,yhalf-0.25,message2,COLOR=8.

 ; Run M.Teig's routine to reset the plotting parameters
 reset_plots,0,0,1

return
end
```

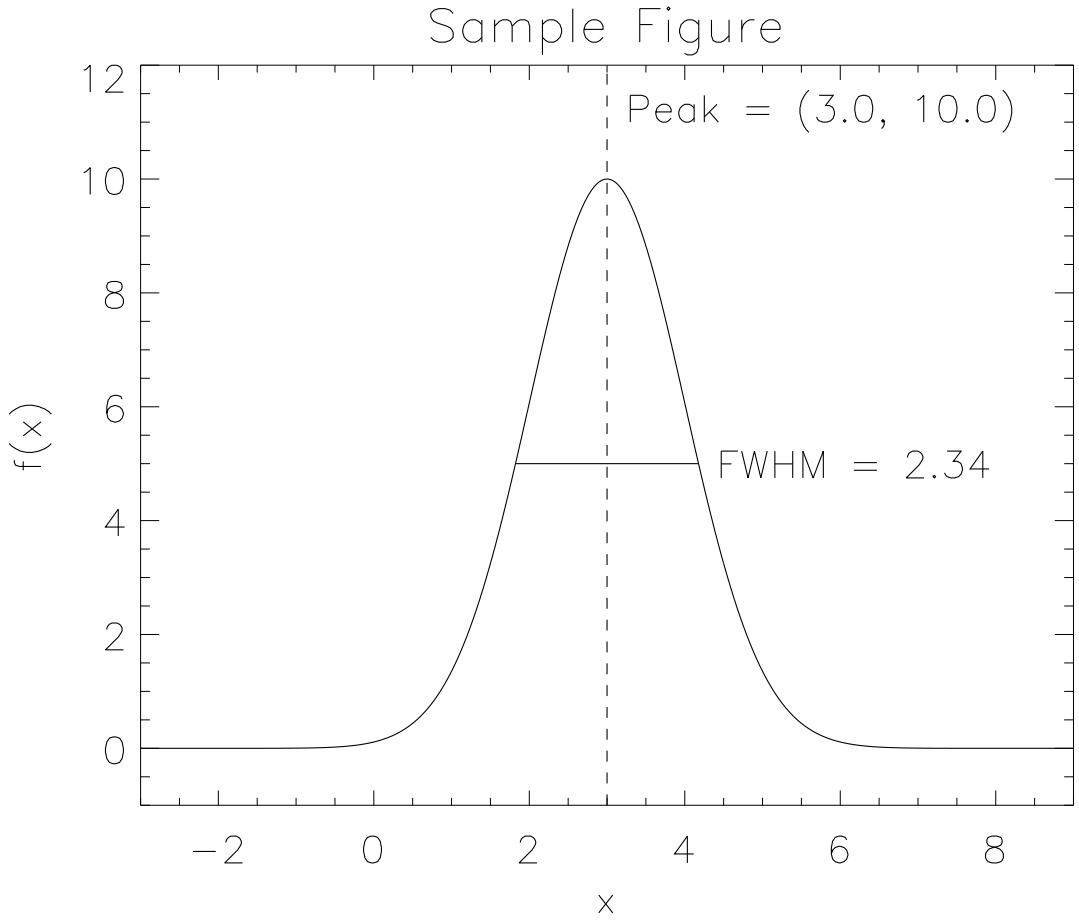

Figure 5: The figure created by the IDL routine in Appendix C.

D Postscripts in IDL

The following page shows how to generate a postscript plot in IDL that you can include in your lab report.

```
; create_hardcopy_1.pro
;
; This is an IDL script to create a hardcopy postscript
; output file called myplot.ps for inclusion in a lab report
;
; Usage:
;   IDL> @create_hardcopy_1.pro

 ; Open the postscript file for output and set up parameters
 set_plot,'ps'
 device,/color,/encapsulated,filename='myplot.ps'

 ; Plot sizes and offsets are given in centimeters
 device,xsize=18.,ysize=24.,xoffset=0.,yoffset=0.

 ; Set black to be the default plot color
 !p.color=8

 ; Now run whatever macro produced the plot
 plot_orbit,[0.,25.,45.,65.,90.]

 ; Now close the file and return to plotting in the X window
 ; with black as the default color
 device,/close
 set_plot,'x'
 !p.color=8

 ; Display the plot with the Linux routine gv (run in
 ; the background)
 $ gv myplot.ps &

 ; Convert the postscript to a jpeg file suitable for
 ; incorporating into a Powerpoint or MSWord document
 $ convert myplot.ps myplot.jpg
```

E IDL's CURVEFIT Routine

The following pages show the help page for the IDL routine CURVEFIT.