Deep R-Band Surface Photometry of NGC891

An Honors Thesis

Presented to the Physics Department

of Oberlin College

by

Eric Miller

April 1996
Acknowledgments

Foremost thanks go to Dan Stinebring and Heather Morrison for their inspiring guidance and remarkable insight throughout this project. The knowledge they have imparted to me is beyond measure, and their help in times of distress was very much appreciated. Many thanks as well go to Paul Harding for his very important contributions to the project, not the least of which are the beautiful color plots he supplied us with.

Thanks to all the physics majors who have been there to help and support me over the last four years in my quest for higher education. Special thanks to Bryce Denney, Ed Myers, and Paul Voyles for their \LaTeX hints and ready wit.

Many thanks to my parents, without whose support (financial and otherwise) I would not have been able to be writing this.

Thanks also to Antonin Dvořák and Jethro Tull, whose melodies have inspired many sections of this thesis.

Last but certainly not least, special thanks go to Michelle, who helped me through the difficult times when nothing else did. This is dedicated to her.
Table of Contents

1 Introduction ................................. 1
   1.1 Statement of the Problem .................... 1
   1.2 Overview .................................... 2

2 Galactic Structure ............................ 3
   2.1 Stellar Populations ......................... 4
   2.2 Thin Disk .................................... 5
   2.3 Bulge .................................... 5
   2.4 Thick Disk .................................... 6
   2.5 Halo .................................... 7

3 Data and Observations ........................ 9
   3.1 Telescope .................................... 9
   3.2 Detector .................................... 9
   3.3 Data .................................... 11
   3.4 Overview of Data Reduction and Analysis .... 12

4 Preliminary Image Reduction .................... 13
   4.1 Sources of Noise and Unwanted Signal ...... 13
      4.1.1 Photon Noise ............................. 14
      4.1.2 Readout Noise ............................. 14
      4.1.3 Radiation Events .......................... 14
   4.2 Bias Subtraction ............................. 15
   4.3 Dark Current ............................... 15
   4.4 Fixing Bad Pixels ............................ 17

5 Flat-Fielding .................................. 19
6 Final Galaxy Image .......................................................... 31
   6.1 Extinction .................................................................. 31
   6.2 Shifting Frames .......................................................... 32
   6.3 Combining the Galaxy Frames ...................................... 33
   6.4 Rotation .................................................................. 33

7 Calibration to the Standard System .................................... 35

8 Masking ........................................................................ 41
   8.1 Dust Lane Masking ....................................................... 42
   8.2 Star Masking .............................................................. 42
      8.2.1 Finding the Stars .................................................. 43
      8.2.2 Aperture Photometry ........................................... 44
      8.2.3 Constructing the Point Spread Function (PSF) ........ 44
      8.2.4 PSF Photometry ................................................... 47
      8.2.5 Making the Star Mask ......................................... 49
   8.3 The Final Masked Image .............................................. 50

9 Results ........................................................................... 53
   9.1 One-Dimensional Modeling ......................................... 53
      9.1.1 Binning the Data .................................................. 53
      9.1.2 Error Modeling .................................................... 54
      9.1.3 Perpendicular Profiles ......................................... 57

10 Conclusions .................................................................. 65
   10.1 Analysis of Results .................................................... 65
   10.2 Further Analysis ........................................................ 66

A Glossary ......................................................................... 67

B Aperture Photometry ........................................................ 71

C Program Listings .............................................................. 75
   C.1 newmode .................................................................. 75
   C.2 spikemode ............................................................... 80
   C.3 radial ..................................................................... 85
   C.4 mask ...................................................................... 88
   C.5 profile .................................................................... 92
   C.6 magnitude .............................................................. 103

References ......................................................................... 105
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>The Burrell-Schmidt telescope at Kitt Peak National Observatory</td>
<td>10</td>
</tr>
<tr>
<td>4.1</td>
<td>Sample CCD image</td>
<td>16</td>
</tr>
<tr>
<td>5.1</td>
<td>Modal sky value vs. time of observation for all four nights</td>
<td>21</td>
</tr>
<tr>
<td>5.2</td>
<td>Brightness of night 4 stars</td>
<td>24</td>
</tr>
<tr>
<td>5.3</td>
<td>Histogram of bias-corrected night 1 sky flat</td>
<td>25</td>
</tr>
<tr>
<td>5.4</td>
<td>Histogram of median-smoothed night 1 sky flat</td>
<td>26</td>
</tr>
<tr>
<td>5.5</td>
<td>First-order super sky flat</td>
<td>28</td>
</tr>
<tr>
<td>5.6</td>
<td>Histograms of unflattened and flattened sky flat image</td>
<td>29</td>
</tr>
<tr>
<td>6.1</td>
<td>Combined final galaxy image</td>
<td>34</td>
</tr>
<tr>
<td>7.1</td>
<td>Zero-point magnitude calibration</td>
<td>37</td>
</tr>
<tr>
<td>8.1</td>
<td>Fitting a point spread function to a star</td>
<td>46</td>
</tr>
<tr>
<td>8.2</td>
<td>Final point spread function</td>
<td>48</td>
</tr>
<tr>
<td>8.3</td>
<td>Masked final galaxy image</td>
<td>51</td>
</tr>
<tr>
<td>9.1</td>
<td>Positions of the centroids of data bins</td>
<td>59</td>
</tr>
<tr>
<td>9.2</td>
<td>Profiles parallel to the minor axis</td>
<td>60</td>
</tr>
<tr>
<td>9.3</td>
<td>Folded profiles for three vertical strips</td>
<td>61</td>
</tr>
<tr>
<td>9.4</td>
<td>Fitted perpendicular profiles</td>
<td>62</td>
</tr>
<tr>
<td>B.1</td>
<td>Typical stellar profile used for aperture photometry</td>
<td>73</td>
</tr>
</tbody>
</table>
List of Tables

5.1 Magnitudes of stars for all four nights ....................................... 23

7.1 Key to color plot of NGC 891 ....................................................... 39

9.1 Table of results ........................................................................... 63

10.1 Comparison of results with previous research .............................. 66
Chapter 1

Introduction

1.1 Statement of the Problem

Compared to the estimated lifetime of a typical galaxy, the scant century of galactic observation is a brief span of time. This poses a problem for astronomers who wish to understand the formation and evolution of these stellar systems. Since it is virtually impossible to view a galaxy as it is forming, we are forced to find other means of determining its initial conditions. Fortunately, there are billions of observable galaxies of different types in various stages of evolution. A comparative handful of these are close enough to allow detailed study, but even these contain a wealth of information. Similarities and differences in structure among different galaxies can give important clues about their history.

Morrison et al. (1994) point out that it makes sense to study the oldest populations of stars in a galaxy in order to extrapolate to the conditions at the birth of the galaxy. As we shall discuss in Chapter 2, the oldest stars are normally found above and below the plane of a spiral galaxy. Much work has been done on this in the Milky Way; however, to get a good sample, external galaxies need to be studied as well. Spiral galaxies which appear edge-on\(^1\) to us are the most informative to study, since the important regions can be observed independently of stars in the spiral arms.

The major problem inherent in such observations is the low surface brightness of these

\(^1\)All *italicized* phrases in this thesis refer to terms that are defined in the Glossary (Appendix A).
galactic components. In the Milky Way it is possible to resolve and count individual stars; in
most external galaxies this is not the case. For this reason, Morrison et al. (1994) developed a
new method of data reduction and analysis that takes advantage of the highly linear response
of the CCD camera. Their method, which is used in this thesis, also requires taking as much
calibration data as object data and demands extensive error modeling to maximize the depth
of observation and ensure that what is observed is real.

1.2 Overview

The work presented in this thesis represents an effort to further capitalize on this new
method of data analysis. We have followed much of the procedure outlined by Morrison
et al. (1994) for the edge-on spiral galaxy NGC 891. After a brief treatment of galactic
structure and formation theories (Chapter 2) and a description of the observations (Chapter
3), we will examine the data reduction steps in detail including the preliminary reduction
steps (Chapter 4), production of a high-quality flat-fielding calibration image (Chapter 5),
creation of a clean final galaxy image (Chapter 6), and construction of a mask to remove all
light that does not originate in the galaxy itself (Chapter 8). Calibration of our data to the
standard system will also be discussed (Chapter 7). Finally, the results will be modeled and
discussed (Chapters 9 and 10).

To be able to analyze the character of these faint outer galactic regions, we will need to
observe light which is 8 stellar magnitudes (or 1500 times) fainter than the background light
of the night sky. We will be exploring new regions of this particular galaxy to find clues
about its formation. The success of this effort will hinge on an accurate characterization of
the instrumental response function and careful attention to error analysis.
Chapter 2

Galactic Structure

A spiral galaxy such as the Milky Way (hereafter “the Galaxy”) is thought to be made up of several components. All spiral galaxies are characterized by a rotating thin disk of stars, also referred to as the “spiral arms”. In addition, spiral galaxies may contain a bulge of various shapes and sizes, as well as thick disk and halo components.

The galaxy we have studied, NGC 891, is a spiral galaxy 9.5 Megaparsecs (Mpc, or \(10^6\) parsecs) distant in the constellation of Andromeda. Long considered the Milky Way’s twin, NGC 891 has an appreciable bulge and dust lane. Its apparent similarity to the Milky Way makes it a very tempting subject for research. In addition, the galaxy is positioned nearly edge on (i.e. with its rotational axis perpendicular to our line of sight), allowing us to examine the brightness variation as a function of height above and below its galactic plane.

To provide motivation for this thesis, some understanding is needed of current theories of galactic structure and questions they raise about galactic formation. The remainder of this chapter will summarize the character of the major components of large spiral galaxies and will bring up issues of active research. The Milky Way will be used as an example, since it is the most studied and well-understood galaxy. Despite its apparent similarities to NGC 891, however, it should not be assumed to be perfectly representative of all spiral galaxies.
2.1 Stellar Populations

Historically, stars have been grouped into two distinct populations. So-called Population II stars are mostly very old, red dwarf stars. The oldest of these formed 15 billion years ago, around the time the Galaxy itself was condensing out of the intergalactic medium. Population I stars are characteristically younger stars, such as our Sun, although they have a fairly wide range of ages compared to Population II.

A key feature that distinguishes the two populations is their metallicity, or metal\(^1\) content. During the formative era of the universe, elements heavier than helium were extremely scarce. As massive stars began to evolve and explode in supernovae, the metals that had been produced in their cores, as well as metals created in the explosions themselves, were strewn throughout interstellar space. As later generations of stars condensed, their composition included a richer abundance of elements such as carbon, oxygen, nitrogen, silicon, and iron. Thus the oldest Population II stars have very low metallicities, while the younger Population I stars have relatively high metallicities. Evidence of this can be seen in the relative abundance of heavy elements in the solar system, the neighborhood of a Population I star.

The relative abundance of each population in various galactic locations says something about the evolutionary history of the Galaxy. Population II stars are found in the bulge and halo of our galaxy (described below), while younger Population I stars are found in abundance in the spiral arms. As a rough approximation, this would imply that the halo and bulge condensed earliest, with the disk evolving at a later date. Recent evidence suggests that there are not simply two populations but a wide variety of stars with different ages and metallicities. (See, for example, Preston et al. 1994; Hartkopf and Yoss 1982.) By analyzing the location and dynamics of various populations, we can infer many things about the structure and formation of the Galaxy.

\(^1\)In astronomical terms, "metal" refers to any element heavier than helium; it is a useful generalization since the overall composition of the universe is thought to be about 90% hydrogen and 10% helium by number, with all other elements making up the minute difference.
2.2 Thin Disk

The thin disk component contains most of the luminous matter of a spiral galaxy, \( \sim 6 \times 10^{10} M_\odot \) in the Milky Way. \((M_\odot \) refers to a solar mass, or \( 2.0 \times 10^{30} \) kg.) Gas and dust abound in the thin disk, giving rise to regions of star formation. As a result, the stars in the thin disk are mostly young, metal-rich, Population I stars. The scale height of the Galaxy’s thin disk is of order 300 pc; the scale length is roughly 4–5 kpc (Freeman 1987). The density relation for the thin disk is exponential, given by

\[
\rho(R, z) \propto e^{-R/h_R - |z|h_z},
\]

where \( h_R \) is the scale length, \( h_z \) is the scale height, and \( R \) and \( z \) are radius and height in a cylindrical coordinate system (Freeman 1987).

The thin disk scale height for NGC 891 has been measured to be about 0.5 kpc (van der Kruit and Searle 1981b). The brightness distribution function used in this determination is of the form

\[
\rho(z) \propto \text{sech}^2 \left( \frac{z}{z_0} \right).
\]

At large \( z \), the factor \( z_0 \) becomes twice the exponential scale height \( h_z \). The measured scale height is probably somewhat inflated, for reasons which will be described in Chapter 9 of this thesis.

2.3 Bulge

Most large spiral galaxies contain bulges in one form or another. The Galaxy’s central bulge contains \( \sim 10^{10} M_\odot \) (Morrison 1996), in the form of old, red stars (Population II). There is virtually no star formation in this region.

There is increasing evidence that our galaxy might have a “boxy” bulge, that is a bulge which appears rectangular in cross-section. This could come about from rotational flattening and could indicate that our galaxy is a barred spiral, rather than a spiral with a spherically symmetric bulge.

For this project, we will not be analyzing much of the bulge; in fact, we will end up masking out most of its light.
2.4 Thick Disk

The Galaxy’s thick disk exists between the massive thin disk and the faint extended halo. Its scale height is $h_z \approx 1.0 - 1.3$ kpc, several times the scale height of the thin disk (Gilmore and Reid 1983; Norris 1996); however, the mass density of the thick disk at $z \approx 0$ has been estimated at only 2%-5% that of the thin disk (Norris 1996). This number is known as the zero-height (or local) normalization. The scale length of the Galaxy’s thick disk has not been reliably measured.

The thick disk contains stars of an intermediate population: Hartkopf and Yoss (1982) found that they are not as metal-rich as thin disk stars, but contain more metals than stars associated with the extended halo. In addition, the rotational velocities of thick disk stars fall between the rapid rotation of the thin disk and the slow, dispersed rotation of the halo (Freeman 1987).

Several major questions about the structure and origin of the thick disk still remain. Are the thin and thick disks discrete components, or is the thick disk just an extended, energetic part of the thin disk? Did the thick disk evolve before the thin disk when the Galaxy condensed out of the intergalactic medium, as a “top-down” model such as that of Sandage and Fouts (1987) would predict? Or did the thick disk form from later expansion of the thin disk due to heating? If so, was the heating mechanism gradual or more rapid, such as would be caused by the accretion and frictional deceleration of a satellite galaxy (Freeman 1987)? Such satellite accretion has recently been observed in the Milky Way (Ibata et al. 1994). Evidence for a history of this process has been put forth by Preston et al. (1994), who discovered a population of young, blue, metal-poor stars in the Galactic halo. These stars were most likely stripped from a companion dwarf galaxy sometime in the past $10^{10}$ years.

Recent evidence also suggests that thick disks and bulges are linked. Van der Kruit and Searle (1981a) found that there was no apparent thick disk for edge-on spirals with a negligible bulge. This has since been confirmed for NGC 5907 by Morrison et al. (1994), who went significantly fainter than the original photographic plate photometry. Likewise, edge-on galaxies with more significant bulges have been found to have a thick disk component. “Top-down” thick disk formation models predict that every spiral galaxy should have a thick
disk. This result would hence seem to favor models in which the thick disk forms at a later epoch than the thin disk, such as by satellite accretion.

One galaxy that has been observed to have a thick disk is NGC 891, analyzed by van der Kruit and Searle (1981b) and Bahcall and Kylafis (1985) using the same photographic plate data. This project hopes to expand on that work to a level several magnitudes fainter, using the linearity and well-defined noise parameters of a CCD detector to their full advantage.

2.5 Halo

The Galactic halo contains about $10^9 \ M_\odot$ of luminous mass, a relatively insignificant amount of the total mass of the Galaxy. The majority of halo stars are Population II, indicating that stellar formation ended long ago and all that remain are old, red stars born in the Galaxy’s youth. Notable constituents of the halo include the globular clusters, highly concentrated collections of $10^4$–$10^6$ extremely metal-weak stars. These are thought to be the oldest objects in the Galaxy, around 15 billion years old. There are roughly 100 globular clusters in the Galactic halo.

The halo is believed to have formed the earliest of the Galactic components; this idea is supported by the character of the stellar halo population. It is interesting to note that populations further from the plane of the Galaxy have lower metallicities and higher velocity dispersions, and are thus thought to be older (Freeman 1987).

There are many ideas about the distribution of matter in the Galactic halo. Various studies have measured spherical density distributions similar to $\rho \propto r^{-3.5}$ (Zinn 1985; other work summarized by Majewski 1993). The local normalization seems to be about 0.2%. There is significant disagreement about whether the halo is flattened, and to what degree. It is also unclear whether the stars and clusters observed in the halo are the primary mass contributors or whether they simply trace the potential of a much fainter (or dark) matter distribution. Evidence for the latter was put forth by Sacket et al. (1994), who observed a faint luminous halo around NGC 5907 using the same reduced data as Morrison et al. (1994). They determined an approximate $r^{-2}$ density distribution, much less steep than the usual $r^{-3.5}$ luminous halo distribution, and concluded they were observing stars which
traced the potential of the *dark matter* halo. The density distribution of the dark halo has been derived from rotation curves, and fits well with their data.

In this project, we hope to observe faint enough to detect the presence (or lack of) a luminous halo. Whether we do so or not will depend on how well we can reduce image calibration errors. As we shall see, these have the largest effect on the most extended galaxy light.
Chapter 3

Data and Observations

3.1 Telescope

The data for this project were obtained in October 1994 by Heather Morrison (Case Western Reserve University), using the 0.61m Burrell Schmidt\(^1\) reflector at the Kitt Peak National Observatory\(^2\). This telescope, shown in Figure 3.1, has a focal ratio of f/3.5, an aperture 24” in diameter, and a spherical primary mirror 36” in diameter. These factors give it a fairly wide field of view; the attached CCD camera (described below) images nearly 70 arcmin of the sky. This is a good size for observing the extended light of an edge-on spiral galaxy; NGC 891 subtends an angle of about 14 arcmin.

3.2 Detector

The Burrell Schmidt is equipped with a CCD camera containing a Tektronix 2048×2048 chip. The silicon CCD chip is an array of photosensitive pixels, each 21 μm in size. When exposed to light, each pixel registers a certain number of electrons produced by the photoelectric effect. At the end of the exposure, the electrons are read out of the chip and converted

---

\(^1\)The Burrell Schmidt telescope is owned by the Warner and Swasey Observatory, Case Western Reserve University.

\(^2\)Kitt Peak National Observatory (KPNO) is a division of the National Optical Astronomy Observatories (NOAO), which are operated by the Association of Universities for Research in Astronomy, Inc. (AURA) under cooperative agreement with the National Science Foundation.
Figure 3.1 The Burrell-Schmidt telescope, owned by Case Western Reserve University and located at Kitt Peak National Observatory near Tucson, Arizona. Image courtesy of Warner and Swasey Observatory Library, http://www.cwru.edu/CWRU/UL/AS/WSOhome.html. (Web page maintained by William Claspy.)
into ADU, or analog-to-digital units. This data is stored in binary form, allowing accurate computer reduction and analysis.

One of the major advantages of CCD detectors is their linearity. The number of ADU reported for a given pixel is directly proportional to the number of photons detected by that pixel. The chip with which our data was taken has an upper limit of 150,000 electrons before the linearity degrades appreciably (by 0.1%). At the gain level that was used (2.5 electrons/ADU), the maximum good data value was actually limited by the 16-bit analog-to-digital converter. This device has an upper limit of 32,767 ADU, corresponding to a reliable upper limit of 82,000 electrons. This point is called the saturation point; any pixel which registers near 32,767 ADU in raw data is probably saturated and must be excluded from analysis.

The camera is equipped with a set of UBVRI Harris filters. For this work, the R (red) filter was used. It has peak transmission (∼90%) at a wavelength of 5900 Å. The quantum efficiency of the chip at this wavelength is roughly 40%.

The image frame covers 69.2 arcmin of the sky on a side; this corresponds to a pixel size of 2.03 arcsec. The chip is cooled by a liquid nitrogen dewar to a temperature around -100°C. At this temperature, the dark current (see Section 5.3) is basically negligible. At temperatures much lower than this, the charge transfer capability of the silicon degrades significantly (Buil 1991, p. 38).

A typical CCD image is shown in Figure 4.1. The various parts of the image, as well as the various image types, will be discussed in the next chapter.

### 3.3 Data

Observations consisted of 231 R-band images, each 2048 × 2048 pixels in size. The following images, listed with their integration times, were obtained:

- 97 bias frames (0 sec)
- 31 dark frames (600 sec)
- 13 standard star field frames (20 sec)
- 45 sky flat calibration frames (600 sec)
- 45 galaxy frames (600 sec)
The images were taken during the nights of October 27–31, 1994\(^3\). One of the five nights was clouded out; observations for the other four were made in apparently \textit{photometric} conditions.

### 3.4 Overview of Data Reduction and Analysis

The majority of work done for this thesis involved careful reduction of the data. Because we wished to analyze extremely faint light, it was necessary to be as accurate as possible on each of the reduction and calibration steps. The remaining chapters will examine this process in detail.

Most of the image processing performed for this thesis was done using \textsc{IRAF}\(^4\). In addition, several programs written by Dr. Morrison were used for reduction and analysis. Complete code listings of these programs are provided in Appendix C for future reference.

---

\(^3\)Dates listed are at the beginning of the run, Mountain Standard Time.

\(^4\)\textsc{IRAF} is distributed by the National Optical Astronomy Observatories (see Footnote 2).
Chapter 4

Preliminary Image Reduction

The initial steps of image reduction remove additive effects which arise solely from the CCD camera and electronics. These effects are present in all images, but they are not necessarily constant over time.

In this chapter, we will first consider the sources and magnitude of noise present in a CCD image. With this in mind, we will then examine the most important preliminary steps of data reduction.

4.1 Sources of Noise and Unwanted Signal

Each pixel in a CCD image has a certain level of random noise associated with it. This noise comes from several different sources, most of which are independent. Two of these sources of noise, photon statistics and readout of the chip, will be mentioned in this section. Other sources, such as sensitivity variations of the chip and background sky fluctuations, will be dealt with in later chapters.

Random noise can be reduced by combining many images together. For an image with a signal $C$ and a noise level $\sigma_C$, the signal-to-noise ratio is given by $C/\sigma_C$. Using the mean of $n$ images will improve this by a factor of $\sqrt{n}$; using the median will improve it slightly less, by a factor of $\sqrt{n}/1.22$. (This additional factor of 1.22 was checked by Morrison et al. 1994).
4.1.1 Photon Noise

Photon arrival statistics are Poisson distributed, with a signal of $N$ photons having $\sigma_p = \sqrt{N}$. The CCD chips detects a certain fraction of the photons that strike it, determined by its quantum efficiency ($\sim 40\%$ for this chip in R). The electrons produced by these photons are read out and converted into ADU by the camera electronics. The photon noise in ADU is $\sigma_{p,ADU} = \sqrt{gC}$, where $g$ is the gain (in electrons/ADU) and $C$ is the number of ADU, also called the number of “counts”. Note that the gain for our system was 2.5 electrons/ADU.

4.1.2 Readout Noise

Some amount of random noise is introduced when the chip columns are read out, amplified, and fed through the analog-to-digital converter. This noise is independent of exposure time, and is fairly well characterized for an individual CCD chip. For the chip we used, the readout noise was 3.0 electrons, or 1.2 ADU.

4.1.3 Radiation Events

Other features that can appear in an image are not really random noise, but one-time events. Cosmic rays are examples of such events; other events are caused by radiation emanating from within the chip itself. Together, these are called radiation events, although they are often indiscriminately referred to as cosmic rays. Any cosmic ray incident on the chip during an exposure has a chance of being detected; the number of such detections is proportional to the integration time. With our integration time of 600 s, there were a significant number of cosmic ray events in each frame. In addition, most cosmic ray detections are indistinguishable from faint stars, which poses a bit of a problem.

The easiest way to remove cosmic rays and other radiation events is to combine several images together, throwing out the maximum pixels in each group to be combined. It is rare that a cosmic ray hits the same pixel twice in a set of images. This maximum value clipping is especially important for combining algorithms which average the combined images; for an algorithm which finds the median of the images, it is still a useful thing to do.
4.2 Bias Subtraction

Any image taken with a CCD camera has a bias level associated with each of its pixels. This bias is a measure of the zero-point offset, or “pedestal” value of that pixel. The bias level does not necessarily remain constant from image to image, although the general bias structure of the chip does. To remove this additive bias level, two types of reduction are necessary. One of these accounts for the magnitude and variation in the level of the zero-point during the night. The other accounts for the overall structure of the bias level across the chip.

The first step in reduction is to apply the overscan correction. The overscan region is an extra section of the chip which does not get illuminated; for our data, it was read out as columns 2049–2080 (see Figure 4.1). To apply the correction, the region was averaged along each row, then a constant function was fit to this average column. This function was subtracted from each column in the data section to remove the bias for that image. The typical bias level was approximately 650 ADU.

The second step in bias correction is to account for the differences in bias level as a function of position on the chip. This is done by using a “master” bias frame. A bias frame is a zero-time integration which measures only the bias level of the chip. About twenty such frames had been taken each night of observing. These twenty frames were combined into a “master” bias image to reduce the effect of readout noise. The master bias for each night was then subtracted from each image taken that night.

4.3 Dark Current

Dark current is caused by thermal vibration of the CCD chip’s silicon lattice. Even when there is no light entering the camera, the chip pixels still register a number of electrons, and thus counts, due to this effect. The magnitude of the dark current is sensitively dependent on temperature; the dark current signal $C_D$ is given by

$$C_D \propto T^{3/2} e^{-V_g e / 2kT}$$  \hspace{1cm} (4.1)

where $T$ is the temperature in Kelvins, $V_g$ is the voltage gap of the semi-conducting material,
Figure 4.1 A sample image showing the typical parts of a CCD image. Rows are horizontal strips ranging from 1 to 2048, and columns are vertical strips ranging from 1 to 2080. Columns 2049 to 2080 are the overscan region; they are not exposed to light, only read out to obtain the bias level of the image. Note the two bad columns in this image; these are cosmetic defects of the actual chip, thus they appear on every image taken with this CCD.
$q_e$ is the charge of an electron, and $k$ is Boltzmann’s constant (Buil 1991, p. 38). Associated with this dark current is a thermal noise related to $\sqrt{C_D}$.

To account for the dark current, images called dark frames (or simply darks) are taken. These are images with the camera shutter closed, taken for the same integration time as the object images to which they will be applied. Several darks are combined to create a master dark frame, similar to the master bias frame; this step reduces the effects of the thermal noise and readout noise when the dark current is subtracted from each object image.

The CCD chip used in our observations is liquid nitrogen cooled to a temperature of nearly -100°C. The quality of the chip is such that at this temperature the dark current is negligible. Several dark frames had been taken during observations. The bias correction was applied to these frames and the dark current was found to be extremely low. We thought it better not to apply this correction rather than introduce extra noise into each image.

### 4.4 Fixing Bad Pixels

The final step in preliminary reduction is to get rid of the bad parts of the image. The bad parts are regions on the chip which are insensitive to light (bad pixels) or parts of the image which the chip is unable to read out for some reason (bad columns; see Figure 4.1). The measured counts in these regions will not be zero due to the addition of readout noise; however, we certainly do not wish to include them in a procedure which does something like taking the mean of the image.

The bad pixels can be flagged using a bad pixel mask within IRAF. This mask is simply a list of the bad pixels in the frame. Any tasks run in IRAF will consult the bad pixel mask and either ignore these pixels or set them to some value which separates them from the good parts of the frame. For convenience, the bad pixels in each of our frames were set to zero.

The edges of the frame are also sometimes questionable in quality. For our data, the last three columns of the data section were found to be rather poor. These were trimmed, along with the overscan region, which was no longer of any use. The removal of the last 35 columns created a final image size of 2045 × 2048 pixels.
Chapter 5

Flat-Fielding

Flat fields are images taken with the CCD chip uniformly illuminated by a fairly bright source. They are used to correct images for systematic brightness variations, such as pixel-to-pixel sensitivity variations, dust shadows on the chip, and optical effects of the telescope. This is accomplished by dividing the data image by a normalized flat field after all the images have been corrected for bias and dark current.

The flat fields taken for this project were sky flats. These are images of regions relatively free of stars, taken for the same integration time and through the same filter as the object images. Because of this, they closely mimic the brightness pattern and variation of the object region, thereby making extremely good flat fields.

The major drawback of using sky flats is that the sky background level is often not bright enough to provide a good signal-to-noise ratio. In addition, there will be stars in the field. To improve results, many sky flats must be imaged and combined to create a master flat field. This “super sky flat” must contain no leftover star features and must have a reasonably high signal-to-noise ratio to achieve a relatively clean final image.

Forty-five sky flats had been taken over the four night observation run. The flats were taken throughout the night in regions nearby the galaxy so that they would accurately reflect the background brightness level and variation observed in the galaxy frames themselves. Between successive flats, the telescope was moved about 10 arcminutes so that each flat was
centered on a slightly different part of the sky. This technique, called \textit{dithering}, prevents bright stars from falling on the same pixels in several flats, an event that could create unwanted stellar “ghosts” in the final super sky flat.

The procedure for creation of the super sky flat was an iterative one; several procedures were attempted, tested, improved, and attempted again. To allow testing of each procedure and for later error analysis, the collection of sky flats from all four nights was split in half. The procedure was performed on each set independently, then one resultant super sky flat was divided into the other to assess the quality of the calibration. This can be thought of as correcting one super sky flat with the other. The ideal resultant image would look flat over the whole frame, containing only random noise from photon statistics and readout noise. Any large-scale sensitivity variations would be removed. The procedure we used is described below.

First each sky flat was inspected for scattered light and extremely bright stars or clusters of stars. One sky frame was found to be undesirable and was rejected, leaving a total of 44 acceptable frames, in groups of 21 and 23.

Next, each sky flat was put through a program called \texttt{newmode} written by Dr. Morrison to accurately determine the mode of the frame’s pixel values. This number, representing the most common brightness value in the frame, should correspond to the sky background level. The results of this procedure were unexpected in two respects. First, there was a marked and systematic variation in the sky brightness among separate frames, ranging from a value around 1600 ADU to a value over 2700 ADU. Furthermore, variations in the mode were discrete, coming in separations of approximately 128 ADU. Figure 5.1, plotting mode of the sky level vs. time of observation for each night, demonstrates both of these effects.

The large, unexpected sky variations implied that conditions might not have been photometric during the entire observation. Very high, thin clouds passing overhead could have caused a significant drop in the amount of light reaching the telescope, changing both the background and stellar brightness levels. To test this hypothesis, \textit{photometry} was performed on ten stars in each of the galaxy frames. (The type of photometry performed, \textit{aperture photometry}, is described in detail in Appendix B). Since it is an additive effect, sky brightness should have no effect on the measured brightness of a star. Non-photometric conditions such
Figure 5.1 A graph of modal sky value vs. time of observation for all sky flats and galaxy frames over all four good nights. Note the magnitude and discrete nature of the variations in background brightness.
as thin clouds are multiplicative effects and would thus alter the amount of light received from a star. The results of the photometry are shown in Table 5.1 for all four nights and are plotted in Figure 5.2 for night 4 stars. Comparison of Figure 5.2 with Figure 5.1 shows that the stellar brightnesses were essentially constant and hence were not varying in the same manner as the sky background. Analysis of Table 5.1 shows that despite the nearly 50% background variation, stellar brightnesses varied by little more than 2% over each night. The few discrepant values (e.g. stars 2 and 3) were probably an effect of the roughness of the photometry performed. Much more accurate photometry was performed later with similar results.

We concluded that the variation was caused by varying emission from molecules in the upper atmosphere. Variation of this scale is common in the I-band (infrared), but is not normally observed in R-band data (King 1995). However, because of the consistency of the stellar magnitudes, we decided the data were still usable.

The discreteness of the modes (see Figure 5.1) was initially quite puzzling. Upon further investigation, we realized that the histogram of each image had “spikes” at regular intervals of pixel value (in ADU). This phenomenon is shown in Figure 5.3 for a sky flat taken the first night. Note that individual spikes appear at intervals of about 32 ADU, while at intervals of 128 ADU (every fourth spike) there is a significantly larger spike. In order to find an accurate mode, newmode chopped off the top ten percent of the histogram, then found the median of these remaining values. In such a spiky histogram, however, the program would only chop off the top of the large spikes. Thus the median would fall on one of the spikes.

The most likely explanation for this is an imperfection in the 16-bit analog-to-digital converter. Each bit in the converter bins up data in a certain range to change it to an integer from 0 to 32,767. The bits corresponding to multiples of 32 and 128 had slightly larger bins than the others, causing more pixels to be read out at these multiples and resulting in spikes in the histogram. While this may appear to be a major problem, we were reassured by the technical staff at Kitt Peak that this is within acceptable parameters. Indeed, the number of pixels affected is rather small. The largest spike in Figure 5.3 (at a pixel value around 1950 ADU) is 3500 pixels above the rest of the histogram. This spike has a width of 2 bins, and thus contains 7000 pixels. In a 2045 × 2048 pixel frame, there are 4.19 ×10^6 pixels, so
<table>
<thead>
<tr>
<th>Star</th>
<th>Night</th>
<th>Magnitude</th>
<th>$\sigma_I/I$</th>
<th>No. Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>18.30 ± 0.02</td>
<td>0.01</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>18.27 ± 0.02</td>
<td>0.02</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>18.26 ± 0.03</td>
<td>0.02</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>18.29 ± 0.02</td>
<td>0.02</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>17.04 ± 0.04</td>
<td>0.04</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>17.00 ± 0.04</td>
<td>0.04</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>17.02 ± 0.03</td>
<td>0.02</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>17.12 ± 0.05</td>
<td>0.05</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>16.82 ± 0.04</td>
<td>0.04</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>16.81 ± 0.09</td>
<td>0.08</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>16.67 ± 0.02</td>
<td>0.02</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>16.81 ± 0.06</td>
<td>0.05</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>18.11 ± 0.01</td>
<td>0.01</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>18.09 ± 0.02</td>
<td>0.02</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>18.04 ± 0.02</td>
<td>0.02</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>18.14 ± 0.01</td>
<td>0.01</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>18.11 ± 0.01</td>
<td>0.01</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>18.10 ± 0.02</td>
<td>0.01</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>18.16 ± 0.03</td>
<td>0.03</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>18.18 ± 0.01</td>
<td>0.01</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>17.62 ± 0.01</td>
<td>0.01</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>17.60 ± 0.01</td>
<td>0.01</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>17.62 ± 0.01</td>
<td>0.01</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>17.67 ± 0.02</td>
<td>0.02</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>17.60 ± 0.01</td>
<td>0.01</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>17.59 ± 0.03</td>
<td>0.03</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>17.40 ± 0.02</td>
<td>0.02</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>17.41 ± 0.02</td>
<td>0.01</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>17.82 ± 0.02</td>
<td>0.02</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>17.81 ± 0.02</td>
<td>0.02</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>17.82 ± 0.02</td>
<td>0.02</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>17.85 ± 0.01</td>
<td>0.01</td>
<td>15</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>18.02 ± 0.01</td>
<td>0.01</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>18.00 ± 0.02</td>
<td>0.02</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>17.93 ± 0.03</td>
<td>0.02</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>17.94 ± 0.02</td>
<td>0.01</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 5.1 The magnitudes of stars are listed with error for all four nights, along with the standard deviation in actual brightness and the number of images taken each night. The error quoted is based on observations of a particular star throughout the night. The approximate error of no more than 2% in brightness indicates photometric conditions all night for each of the four nights.
Figure 5.2  The magnitudes of selected stars throughout observations during night 4, plotted vs. the time of night. Comparison with Figure 5.1 shows that although the modal sky value varied, the brightness of stars remained constant to within about 2%. This implies an additive variation, such as changes in the sky background level, as opposed to non-photometric conditions. The magnitudes were found from aperture photometry performed in IRAF.
Figure 5.3 Histogram of a night 1 sky flat showing the spikes which resulted in a miscalculation of the mode. The spikes, which were corrected, were most likely caused by an imperfection in the analog-to-digital converter.

This particular spike only contains 0.17% of the pixels in the entire image. The other spikes are significantly smaller than this one, so the total effect is probably no more than 1%.

To obtain a more accurate estimate of the mode, newmode was changed so that it would smooth out the spikes with a median smoothing algorithm. This algorithm considers the values within a window about a point in the histogram. The median of these values is found and substituted for the point at the center of the window, then the window is stepped along the histogram to find the next point. Such an algorithm removes outliers which are significantly narrower than the smoothing window.

The new program, spikemode, used a window of 13 points. Once the histogram was smoothed (see Figure 5.4), spikemode chopped off the top ten percent and found the median of these points. This value was assigned to the mode of that particular image.

With accurate modes determined for each sky flat, we were able to proceed to the next step. Both sets of sky flats were fed to the IRAF task imcombine to produce two first-order super sky flats. The imcombine algorithm consists of two basic steps. First, images are read.
Figure 5.4  Histogram of the sky flat from Figure 5.3 after it had been median-smoothed. Note that the spikes have been removed, allowing a more accurate determination of the mode.

In and scaled by some value, so that they are all clustered around a common point. In this case the modes determined above were used. Second, corresponding pixels on each image are considered and run through a smart rejection algorithm. This algorithm, called `ccdclip`, utilizes the well-defined readout noise and gain values to come up with an accurate value for the expected standard deviation for a given signal $C$ (in ADU), using the formula

$$\sigma = \left[ (N_r/g)^2 + (C/g) \right]^{1/2}$$  \hfill (5.1)

where $N_r$ is the readout noise ($= 3.0$ electrons) and $g$ is the gain ($= 2.5$ electrons/ADU). The first term in this equation comes from readout noise; the second comes from photon noise. These independent variances are then added in quadrature. Corresponding pixels in each of the input images are considered as groups; this is equivalent to stacking the input images on top of each other and grouping all pixels along any vertical line pointed down the stack. The minimum and maximum pixels in each group are thrown out, and the median of the remaining pixels is determined. The expected sigma is calculated from Eq. 5.1, using this median value as $I$. Any pixel more than $2\sigma$ away from the median is rejected from the group, and the procedure is repeated until there are no further rejections. At this point the
median is written to the appropriate pixel in the output image, and the procedure moves on to the next pixel group.

The median is used rather than the average for several reasons. First of all, the median represents real data from one of the sky flats, whereas the average would introduce extra noise due to smoothing. Most importantly, though, the average would be greatly affected by wildly discrepant values, such as if more than one of the input pixels registered a cosmic ray. In addition, because of the many stars in our image frame, it is fairly likely that more than one pixel in a group falls on a star, despite the dithering of the frame between successive sky exposures. A median-centered algorithm is obviously more robust in this situation.

Once the combining procedure was complete, both independent flat fields were normalized. Each flat field was inspected for obvious defects such as stellar ghosts. Then one flat field was divided into the other to measure the quality of the procedure. The resultant flattened image was binned into 50 × 50 pixel sections; this eliminated the small-scale noise from the divided flat field, leaving only the large-scale features. The value of each bin was set to the mean of its 2500 constituent pixels, ignoring the bad pixels in the image. Then the mean and standard deviation of the bins were measured. The mean was very close to one, since the super sky flats had been normalized. The standard deviation was found to be 0.23%. This number shows how well the first-order flat fields corrected large-scale sensitivity variations. It also indicated the quality of our procedure.

An example of a first-order flat field is shown in Figure 5.5. The most striking feature of this surface plot is the large dome-like fall-off. This “dome profile” is an optical effect of the Schmidt telescope design. It appears in every image taken with this telescope/CCD system.

With these first-order super sky flats complete, the second iteration was begun. The sky flat images in each set were divided by their respective (unbinned) flat fields, in essence removing the dome profile and other large-scale sensitivity dependence. This process also narrowed the histograms of the sky flats considerably and removed the spikes caused by the analog-to-digital converter, as can be seen in Figure 5.6. As a result, the mode of each input image could be found much more accurately with the original newmode program, allowing precise scaling by the combining algorithm.

The modes were determined for these flattened sky frames and the images were fed
Figure 5.5 A first-order super sky flat, showing the magnitude of the dome-like vignetting effect, as well as some “dimples” of lower sensitivity. The trenches through the middle and side of the surface plot are bad columns.

through imcombine again. The resultant second-order flat fields were normalized, inspected, divided, and binned to test the reliability of the procedure. The resultant sigma was 0.21%, a 10% improvement over the first-order super sky flats.

The whole process was repeated once more, using the newly refined flat fields to determine an even more precise mode for each sky frame. Sigma for this final procedure was also found to be 0.21%. Further iterations of the procedure did not improve this value, so the process was stopped. The two best super sky flats were retained to be used in later error modeling.

We were now confident that we had the best procedure with which to create a super sky flat. The two original sets of sky flats were combined, and the entire procedure was run on all 44 good sky flats, producing the final super sky flat field. Once the flat field was normalized, we were ready to use it to calibrate our object frames.
Figure 5.6 The histograms are plotted on a logarithmic scale for a raw sky flat and a sky flat which has been calibrated (flattened) by a first-order super sky flat. Flattening has caused the histogram to tighten up considerably and has removed the faulty analog-to-digital converter spikes. The single large spike on the flattened histogram represents the pixels from this image which were placed in the flat field during combining. Also note the rather gradual fall-off to the right of the peak; this is from stars and galaxy light.
Chapter 6

Final Galaxy Image

Once the final super sky flat was complete, it was ready to be applied to the galaxy data. Each of the 45 galaxy images were flattened by the super sky flat, then inspected to make sure they had no unwanted features such as strong scattered light. The flattened galaxy frames passed inspection.

6.1 Extinction

Next the galaxy frames had to be corrected for extinction. Extinction is a systematic reduction in the light reaching the telescope due to the Earth’s atmosphere. It is dependent on the geographical location at which the data are taken, and it is also dependent on the airmass, or the amount of atmosphere through which the light has to travel. A telescope pointed straight up is said to looking through 1 airmass.

Extinction reduces the magnitude of a star by an amount

\[ \Delta m = (\text{extinction})(\text{airmass}) = -2.5 \log \frac{f}{f_0} \]  \hspace{1cm} (6.1)

where extinction is a site- and filter-dependent correction factor, and \( f_0 \) and \( f \) are the true and measured flux, respectively. We are interested in finding the true flux, which we can do by rearranging Equation 6.1 to get

\[ f_0 = 10^{(\text{ext.})(\text{air.})/2.5}(f). \]  \hspace{1cm} (6.2)
Thus all we need to do to correct for extinction is to multiply each of our frames by the factor given above. For Kitt Peak, the extinction correction factor in R is 0.1. Coupled with our maximum airmass value of 1.40, this gives a correction factor of 1.14 for the frame at highest airmass, a fairly significant amount. Most frames were taken around 1.10 airmasses, yielding a typical correction factor of 1.11.

6.2 Shifting Frames

Once the galaxy frames were corrected for extinction, they were ready to be combined. Because of dithering performed at the telescope, each galaxy image was in a different place in the frame. This necessitated shifting each galaxy frame so that the galaxy and stars would line up (register) when the images were stacked on top of each other.

One galaxy frame was chosen to be the reference frame with which all the others would be registered. The image chosen had the galaxy close to the middle of the frame. Fifteen bright, unsaturated, relatively isolated stars from various parts of the frame were selected, and their precise x and y frame coordinates in the reference image were determined. Then the coordinates of these stars were determined in each of the remaining 44 images.

To compute the necessary shift for each image, the IRAF task geomap was used. This task fit a transformation function to the x and y coordinates of each image to be shifted. The function we used was a Chebyshev polynomial, quadratic in x and y, with cross terms allowed. geomap was run interactively, allowing us to examine the residuals of the fit for each image and throw out some of the fitting stars if necessary.

The output of geomap was a database containing the transformation coefficients for each galaxy frame. This database was fed into a second IRAF task, geotran, which actually performed the coordinate transformations. The output from this task was a set of 44 shifted galaxy frames which were now registered with the reference frame.

Before the real combining was performed, the images were summed up to make sure that they actually lined up reasonably well. This summed image was compared with the original reference frame to see whether any of the stars on it had increased in size significantly, a sure sign of misalignment of one of the galaxy frames. The galaxy images were well-matched, and
we were ready to proceed to the final combining step.

### 6.3 Combining the Galaxy Frames

The shifted galaxy frames were combined in a manner similar to the sky flats. First, the mode of each galaxy frame was determined with `newmode`. This number was subtracted from each pixel in the frame to remove the sky background; this was necessary because of the night-to-night variation of the sky brightness.

Next, the galaxy frames were combined using the IRAF task `imcombine`. In contrast to the super sky flat production, no rejection algorithm was used here, since we thought this would adversely affect the photometry in the regions with galaxy light. For the same reason, no scaling was used. The necessity for scaling in the super sky flat production was aggravated by the varying sky background. There was no such variation in the star and galaxy brightness; once the modal sky value was subtracted, the 45 galaxy images were well-clustered.

The galaxy frames were combined with a median operation, as in the super sky flat case, and much for the same reasons. Since stars were no longer a contaminating factor in the combining process, and because of the close clustering of the 45 frames, the median was enough to get rid of radiation events in the final galaxy image.

Once the combined galaxy image was produced, a sky background of 1877.25 ADU was added back on. This value was the mean of the sky values determined by `newmode` for the 45 galaxy frames. It was vital to add it back for later analysis.

The final calibrated, combined galaxy image was now complete. A portion of this clean image is shown in Figure 6.1.

### 6.4 Rotation

To simplify analysis, the final image was rotated so that the major axis of the galaxy (along the galactic plane) was parallel to the x axis of the frame. The angle of the major axis to the horizontal was measured from a detailed contour plot. The angle was found to be 24.8° ± .2°, measured clockwise from the horizontal. The IRAF task `rotate` was used to rotate the final galaxy image by this amount and shift the center of the galaxy to the center of the frame.
Figure 6.1 A portion of the cleaned-up final image of NGC 891. The grayscale used is inverted and logarithmic, to show the faint features of the galaxy.
Chapter 7

Calibration to the Standard System

In order to compare our results to previous work, it was necessary to calibrate our data to the standard system. The brightness values reported by IRAF are based on an arbitrary zero-point. While they are internally consistent, they are completely meaningless when compared to data taken at different times by other telescopes.

The solution to this is to take a number of images of so-called standard stars. These are stars which have been carefully and frequently measured so that their brightnesses are known very accurately. Most of these stars have been observed and compiled by Landolt (1983, 1992), who has identified over a hundred standard star fields, called “Selected Areas”, each containing several dozen standard stars. The measured brightness of these standard stars can be compared with the standard brightness, and a measurement of the magnitude offset can hence be obtained.

Several images of standard star fields had been taken throughout the observations. These images were corrected for bias, trimmed, and flat-fielded in the same manner as the galaxy images. They were also corrected for airmass. Landolt’s (1992) tables were consulted to find stars in these regions which would be possible and appropriate to use as standards. One of the criteria for selection was that the star had to have a color similar to what we would expect for a galaxy viewed edge-on. A color range of $0.7 \leq B - V \leq 1.3$ was adopted; any star within this range was considered a galaxy-colored star. From this subset of stars, only
those which were bright enough to register at least 10,000 ADU at peak, yet not too bright to be saturated, were chosen. This selection process produced five standard stars, two in SA 113 (Selected Area 113) and three in SA 95.

Since we did not really know the color of the galaxy, we needed to obtain some estimate of the error due to color. Four blue stars \((0.0 \leq B - V \leq 0.7)\) and two red stars \((1.3 \leq B - V \leq 2.0)\) were identified using the same brightness criteria as above.

The magnitudes of all nine standard stars were measured for each night they were imaged using aperture photometry within \textit{IRAF}. (See Appendix B.) The standard R magnitudes, calculated from Landolt’s (1992) data, were subtracted from these measured magnitudes to obtain a zero-point magnitude offset. The results of this are plotted in Figure 7.1. The magnitude offset of the galaxy-colored stars was found to be \(9.58 \pm 0.01\) magnitudes; this means that any magnitude reported by \textit{IRAF} was \(9.58\) magnitudes fainter (larger in value) than the actual magnitude. It can be seen that the night-to-night brightness variation of these stars was very small, \(\sim 0.01\) magnitudes or \(1\%\) in brightness. This supports the earlier conclusion that conditions were indeed photometric over the observing time. The color term is \(0.10\) magnitudes over a color range from \(0.0 \leq B - V \leq 2.0\).

Knowing the zero-point offset would be enough if we were only interested in measuring stellar brightnesses. However, we are attempting to analyze the surface brightness of a galaxy. Such a source is significantly more extended than a point-like star.

As a result, we must change our standard calibration result to something more useful. The standard method for measuring surface brightness is in terms of magnitudes per square arcsecond. Thus we wish to ask, “How many R magnitudes per square arcsecond correspond to 1 count per second per pixel?” The answer is somewhat complicated by the awkward logarithmic definition of a stellar magnitude. To find this expression for the zero-point, we measured the total number of counts, \(C\), detected by aperture photometry for a given star. Assuming that all these counts fell on the same pixel\(^1\), and given an exposure time \(t\), we then know that the star has a flux of \(f = C/t\), measured in counts per second per pixel. Any other star with a flux of \(f\), indeed, any pixel with a flux \(f\), would be expected to have

\(^1\)The counts did not fall on the same pixel, but we can simplify things by saying they did, since the area of the aperture cancels out of the equation anyway.
Figure 7.1 The zero-point offset of magnitude is plotted for stars with various $B - V$ colors. Multiple data points for a given color represent observations of a star on more than one night; the night-to-night error is about 1% in brightness. The downward slope towards redder stars shows that the band-pass for this filter does not exactly match that for the standard R filter in the UBVRI photometry system.
the same intrinsic brightness as this star, assuming fairly constant observing conditions.

The intrinsic brightness of the star is determined from its standard R magnitude, $R_{\text{stand}}$. Since the star is assumed to cover one pixel, this corresponds to $R_{\text{stand}}$ mag per pixel. Given that a pixel on this particular CCD chip is $a$ sq. arcsec in size, we are in essence dividing the flux by $a$ to convert to brightness in magnitudes per square arcseconds. This leads us to the formula

$$R_{\text{stand}}(\text{in mag per sq. arcsec}) = R_{\text{stand}} - 2.5 \log \frac{1}{a}$$  \hspace{1cm} (7.1)

for the standard brightness. To find the brightness corresponding to 1 count per pixel per second, we simply divide by the flux. For magnitudes, of course, this is equivalent to adding a term to Equation 7.1:

$$R(\text{in mag per sq. arcsec}) = R_{\text{stand}} - 2.5 \log \frac{1}{a} - 2.5 \log \frac{1}{f}$$  \hspace{1cm} (7.2)

Using a pixel size of $a = 4.12$ sq. arcsec, an exposure time $t = 20$ seconds for the standard star frames, and the appropriate values of $C$ and $R_{\text{stand}}$ for each standard star, we found the zero-point, corresponding to 1 count per second per pixel, to be $R = 22.05 \pm 0.03$ mag per sq. arcsec. For a 600 second galaxy image, this leads to a value of $R = 29.01 \pm 0.03$ mag per sq. arcsec corresponding to 1 count per pixel.

From this, the magnitude brightness of any pixel can be easily calculated. For a pixel with $C$ counts in the galaxy image, the R magnitude is

$$R = 29.01 - 2.5 \log C .$$  \hspace{1cm} (7.3)

The average sky brightness, 1877 ADU per pixel, turned out to be $R_{\text{sky}} = 20.8$ mag per sq. arcsec.

The following page presents a color plot of NGC 891 after subtraction of the sky background. The displayed color levels correspond to ranges of surface brightness, as summarized in Table 7.1.
<table>
<thead>
<tr>
<th>Color</th>
<th>R mag per sq. arcsec</th>
</tr>
</thead>
<tbody>
<tr>
<td>white</td>
<td>20</td>
</tr>
<tr>
<td>purple</td>
<td>20–21</td>
</tr>
<tr>
<td>green</td>
<td>22</td>
</tr>
<tr>
<td>yellow</td>
<td>23</td>
</tr>
<tr>
<td>orange</td>
<td>24</td>
</tr>
<tr>
<td>olive</td>
<td>25</td>
</tr>
<tr>
<td>dark blue</td>
<td>26</td>
</tr>
<tr>
<td>light blue</td>
<td>27</td>
</tr>
<tr>
<td>navy</td>
<td>28–29</td>
</tr>
</tbody>
</table>

Table 7.1 Key to color plot presented on the next page. Color levels have been quantized; the reported magnitude refers to the brightness at the outer edge of the level. Plot produced by Paul Harding (Steward Observatory).
Chapter 8

Masking

With the final galaxy image produced and the standard star calibration complete, the next step was to remove unwanted signal from our final image. Since we were solely interested in the light from the galaxy, “unwanted signal” refers to everything else in the image; this includes light from stars and background galaxies, as well as the effects of the galaxy’s dust lane. Furthermore, because we wanted to go as faint as possible, we needed to get rid of this signal to a very precise level, while still retaining enough data to allow sufficient analysis.

As Morrison et al. (1994) point out, it is not necessary to attempt to remove stars and other features by smoothing them out. Indeed, this would introduce a significant amount of error, not to mention work, and would not allow the accurate analysis we wanted. The solution is simply to mask out all the light that does not come from either the galaxy or the sky background (since it is impossible to remove the latter without adversely affecting the former). Any brightness variation seen in this masked image would be due to an intrinsic galactic brightness variation, assuming a constant background brightness over the whole frame\(^1\).

A mask is most effectively made by creating an image the same size as the final galaxy image. Every pixel in this image which is to be masked out in the galaxy frame would be set to zero; every pixel which is to be kept would be set to one. Multiplication of this

\(^1\)The background brightness is not constant over the frame; this will be one of the factors that limits our photometric analysis.
mask image by the galaxy image would result in the masked galaxy frame for which we were striving.

8.1 Dust Lane Masking

The dust lane of the galaxy was the first to be masked out. This dust lane can be seen easily as a thick bright line (due to the inverted grayscale) running along the major axis of the galaxy in Figure 6.1. The dust in the thin disk of the galaxy obscures starlight; the same effect can be seen in the Milky Way on a clear summer night. Above and below the thin disk, there is no obscuring dust, allowing us to see the brightness variation directly.

To mask out the dust lane, a strip 30 pixels in height running the length of the major axis (450 pixels) was considered. This strip was broken up into 30 pixel segments, and a contour plot of each segment was examined. The points at which the galaxy light “turns over” and begins falling off into the dust lane were noted for each segment. The corresponding regions in the mask image were set to zero. All other pixels in the image were set to one. This image was the dust mask image; when it was multiplied by the final galaxy image, the resulting image would have zeros everywhere there was dust and normal brightness values everywhere else.

8.2 Star Masking

Masking out the stars was a much more involved task than masking out the dust lane, simply because there were about 48,000 stars in the frame. Although a star is virtually a point source, atmospheric and telescope scattering spread out the light from a particular star over a significant number of pixels. The shape of the light distribution of a star as a function of position on the chip is called the stellar profile; an example is shown in Figure B.1. (See Appendix B.)

If all the stars in the frame were fairly isolated from one another, we would not have needed to worry about these profiles. We simply could have performed aperture photometry, making sure the stellar aperture was large enough to collect the majority of starlight. Indeed, this was the case for the standard star fields. Because our field was so crowded, however,
many of the stars had neighbors within the region that would have been included in such an aperture.

Why did we need to worry about the brightnesses of the stars so much if we were just going to be masking them out anyway? We needed to know how far out to mask them to be sure the starlight was removed to the level of accuracy needed to measure the galaxy light. The outer regions (called the wings) of a star do not contain a significant amount of light compared to the inner regions; however, the brightness level in the wings varies much more slowly than in the central regions, so a sizable error in the estimated profile would have drastic consequences on the size of the mask.

In reality, the limiting magnitude was determined by how much starlight we could mask out while still retaining enough of the image for accurate data analysis. After all, if we were to mask out all the light from every star, we would necessarily mask out the entire image. Thus an accurate estimate of the stellar profile became vital for us to determine exactly how faint we could go.

8.2.1 Finding the Stars

First of all, the stars needed to be identified; this was done using daofind (see Appendix B). However, due to the extremely crowded nature of the field, it was vital to identify as many stars as possible, as every undetected star would have to be masked out later by hand. At the same time, it was necessary to minimize the number of noise spikes incorrectly identified as stars, since an overabundance of these would cause unnecessary loss of data.

The parameter which ultimately determines the number of stars found is the detection threshold. To determine the best compromise between maximum star detection and minimum accidental detection, several runs of daofind were performed with various values for the detection threshold, ranging from 4 ADU to 15 ADU. The results of each run were visually inspected by marking a point on the image display where every detection occurred. The best compromise was found for a detection threshold of 6.0 ADU; this is twice the estimated noise sigma of $\sigma_N = 3.0$ ADU in the final galaxy image.

Using a threshold of this size and a full width at half maximum of 2.6 pixels, daofind detected 53,790 stars. Given a $2045 \times 2048$ pixel frame, this corresponds to an average of one
star in every $9 \times 9$ pixel region. This is quite a crowded field! As we will see, this number is a bit higher than the final number of stars remaining after photometry was performed. The reasons for this are discussed below.

### 8.2.2 Aperture Photometry

Once the positions of all the stars were known, it was necessary to perform aperture photometry to obtain a first-order estimate of their brightnesses. Because of its use throughout this thesis, aperture photometry is described in detail in Appendix B.

### 8.2.3 Constructing the Point Spread Function (PSF)

A *point spread function* (PSF) is a function which models the typical stellar profile. It is most commonly used to perform photometry in a crowded field of stars, where aperture photometry is very inaccurate. This form of photometry (*PSF photometry*) is an iterative procedure which fits the point spread function to each star, corrects the image, then fits the remaining star, and so on. The procedure will be discussed in detail in the next section.

To construct the point spread function, the *IRAF/daophot* task *psf* was used. The *psf* algorithm, developed and described in great detail by Stetson (1987), is a hybrid between an analytic and empirical fit. Stars to be fit are chosen interactively. The task fits an analytic Gaussian-type function to the star’s profile; the exact function that is used is chosen as a task parameter. The residuals of this fit are computed and, after interpolation, are written into a *look-up table*. The look-up table is centered on the centroid of the star, and it gives corrections to the analytic function, producing a much more accurate model of the true stellar profile than the Gaussian function alone. Likewise, the use of an analytic function in the first place reduces the error due to interpolation, increasing the accuracy of the look-up table.

To improve the quality of the point spread function, several stars were used in its construction. These stars, called *PSF stars*, had to be bright and uncrowded, much like the stars used in Chapter 5 to test the photometric observing conditions. To maximize our accuracy, we wished to compute the point spread function out to a radius of at least 15 pixels, and
preferably larger. As a result, the PSF stars could not have any bright neighbors in this region.

Stars meeting these criteria were quite scarce in our extremely crowded field. Nevertheless, after thorough searching, several candidate PSF stars were identified. In addition, the IRAF version of psf allowed us to use the unsaturated wings of saturated stars to compute the point spread function. This was ideal for us, since we were most concerned with modeling light in this region of the stellar profile.

psf was run interactively, allowing the user to view the residuals of a candidate PSF star before it was accepted and added to the PSF look-up table. Among other things, this allowed us to tell whether there were any fairly bright neighboring stars with which to be concerned (see Figure 8.1). Neighbors within the PSF radius (15 pixels) were acceptable, as long as they contributed only a small amount of light. Neighbors with any light contribution within the fitting radius (5 pixels) were unacceptable, since their light would affect the shape of the analytic function.

Once a first-order point spread function was constructed with the best PSF stars we could find, it was applied to the neighbors of each PSF star using the daophot task nstar. This task fits the point spread function to all stars in a group simultaneously; in this case, it is the group of neighbors about an individual PSF star. nstar performs a non-linear least-squares fit to the point spread function for each star, weighting each pixel in the fit by the radial distance from the center of the profile and by the readout noise, which is specified as a parameter. There are three parameters to the fit, namely the x and y position and the brightness of each star. To improve efficiency, the values found from aperture photometry are used as first approximations for these parameters. The output of the task is a list of x and y centers and magnitudes for all the PSF star neighbors. This list is used as input to the task substar, which shifts and scales the point spread function for each star. The scaled PSF values are subtracted from the stars, producing a residual image.

After nstar and substar were executed, this image was inspected to ensure that the neighbors were subtracted cleanly. Then a second-order point spread function was constructed from the residual image, using only the PSF stars whose neighbors had been cleanly removed. The procedure was iterated several times until the point spread function was free
Figure 8.1 PSF fitting for a typical star. The top surface plot shows the actual stellar profile of the star. The lower plot shows the residuals from fitting an analytic function out to a radius of 5 pixels (the fitting radius); the vertical display scale has been enlarged from that of the top plot by a factor of 9. Note that faint neighbors become visible within the 15 pixel PSF radius.
from contaminating neighbor stars.

The final point spread function for this stage was constructed by Heather Morrison and Paul Harding (Steward Observatory), who had access to a more recent and improved version of daophot. A total of 10 stars were used to construct the point spread function, including those saturated stars whose wings were used to model the outer regions. The final point spread function is plotted in Figure 8.2. This function represents the best approximation to an actual stellar profile in our final image.

8.2.4 PSF Photometry

With the point spread function constructed, we were ready to accurately determine the magnitudes of every star in the frame. The task we used was allstar, which works in much the same way as nstar. One major difference is that it dynamically determines which stars to consider as a group, removing or adding stars from a given group as it iterates the fitting procedure. The task also automatically produces a residual image.

The output of allstar was a list of x and y centers and magnitudes for each star in the frame. After rejection, there were a total of 47,776 stars with reported coordinates and brightnesses. As it turned out, some of these objects were not stars at all, but were features like noise spikes. This was acceptable, since the alternative would be to raise the detection threshold, thereby forcing us to mask out many more faint stars by hand.

When allstar had been run, the residual image was carefully inspected at various grayscale levels to make sure the point spread function had done a good job of modeling the stars. This was important for two reasons. First, it was necessary to obtain very accurate brightness measurements of each star so that a proper scaling could be worked out for the stellar mask. If the stars had been removed relatively cleanly, it was a good sign that the PSF photometry had calculated their positions and magnitudes accurately. Second, it was necessary to model the profile most accurately in the wings, since that would determine how large a mask would be produced by a star of a given magnitude. If the point spread function did a good job removing light from the wings, we could be confident that it would produce a proper mask size.
Figure 8.2 The final constructed point spread function. The top image shows the PSF as a look-up table; if the analytic function fit the stellar profile perfectly, this plot would be uniform. The function does not do too well in the core, but does a fairly good job of modeling the wings of the profile. The lower surface plot shows the PSF as a profile, merely the sum of the function and the look-up table values at a given radius.
8.2.5 Making the Star Mask

As the next step in creating the star mask, the values of the point spread function were written out as a radial profile. This file, listing pixel values at radii in the profile, was fed into radial, a program written by Dr. Morrison. The program combines the profile values into radial, single-pixel bins, then trims off the top and bottom 10% and finds the average for each bin. The result is a radially symmetric model stellar profile. Our PSF profile had a radius of 15 pixels.

Once this was done, we were ready to create the star mask, using the program mask written by Dr. Morrison. Input to mask consists of the PSF photometry output, the radial PSF profile, the dust mask, and the final galaxy image. The program reads in the values of the point spread function at each point in the profile radius. Then, for a range of magnitudes, it finds the radius at which the expected brightness falls to less than $C_{min}$ counts above the sky. The expected brightness (in ADU) at a given radius can be calculated from magnitudes by

$$C(r) = C_P(r)(10^{(m_P - m)/2.5})$$

where $C_P(r)$ is the value in ADU of the PSF profile at radius $r$, $m_P$ is the magnitude of the PSF profile, and $m$ is the stellar magnitude determined by allstar. The magnitude of our point spread function was 12.51.

The central algorithm of mask determines the radius at which the signal $C$ for a given brightness falls below $C_{min}$. For stars with $m < m_P$, this was something less than 15 pixels. For stars brighter than $m_P$, of which there were several, $R$ was given a value of 30 pixels. Any reported stars which fell in the dust mask region were ignored, since most of these features were due to clumpiness in the dust lane rather than actual stars.

The output of mask was an image containing the mask from each star as well as the dust mask. To get the best compromise between fainter limiting magnitude and more usable data, we varied the value of $C_{min}$ in the range 0.5 to 2.5 ADU. The best value was found to be $C_{min} = 2.0$ ADU, meaning all starlight contributing more than 2 counts above the sky level was masked from the image. Note that this is about 0.1% of the average sky background level. From the standard calibration results, 2.0 ADU in our final image corresponds to 28.26
mag per sq. arcsec. This is the absolute faintest magnitude that we can hope to obtain results for using this particular mask. In reality, our limiting magnitude is somewhat brighter than this.

8.3 The Final Masked Image

The final step in the creation of the mask was to inspect it and mask out background galaxies and undetected stars by hand. Most semi-resolved background galaxies were either ignored by \texttt{daophot} or masked incorrectly by \texttt{mask} because they did not fit the stellar profile. In addition, just as \texttt{daophot} misread a few noise spikes as stars, an occasional true star slipped through. These features, as well as large diffraction spikes, were masked out interactively in \texttt{IRAF}'s \texttt{imedit}. Special attention was paid to regions with significant galaxy light, where the brightness variation was steep enough to have confused \texttt{daophot}'s sky-fitting algorithm.

The final mask was multiplied by the final galaxy frame to create a masked galaxy image. A portion of this image, centered on the galaxy, is shown in Figure 8.3.
Figure 8.3 A portion of the masked final galaxy frame. The grayscale is linear and inverted, making brighter regions of the frame appear darker in this picture. The circular regions are masked out stars and the horizontal strip is the dust lane mask.
Chapter 9

Results

9.1 One-Dimensional Modeling

How does the surface brightness of the galaxy fall off as a function of perpendicular distance \( z \) from the major axis? This is one of the most important clues to the history and current structure of a galaxy. The best way to determine this is to analyze perpendicular brightness profiles at various distances \( R \) from the minor axis.

9.1.1 Binning the Data

To increase the signal-to-noise ratio of the profiles, we followed the lead of Morrison et al. (1994) and divided the data into vertical strips 40 pixels wide. Each strip was divided into bins ranging from 3 pixels in height near the major axis to 99 pixels in height away from the galaxy. To determine the size of the bins and perform the actual binning, a program called profile, written by Dr. Morrison, was adapted for our data. The program read in the final galaxy image and the mask image. The height of the first bin was 3 pixels, centered along the major axis. For each bin above and below the galactic plane profile determined the bin height at which the brightness across the bin would change by 1%. This was done assuming an exponential brightness fall-off with scale height \( h_y \), using the formula

\[
\frac{\Delta I}{I} = e^{\frac{|y_1|}{h_y}} - e^{\frac{|y_2|}{h_y}}. \tag{9.1}
\]
Here \( y_1 \) is simply the height of the outer side of the previous bin and \( y_2 \) is the factor that is increased until \( \Delta I/I = 0.01 \). The bin height, \( |y_2 - y_1| \), was limited to 99 pixels. This procedure ensured that each bin contained minimal brightness variation yet covered enough pixels to maximize signal-to-noise. Note that the coordinate \( y \) we are using refers to the vertical distance in the image frame; for our edge-on galaxy aligned with the horizontal, this is identical to the \( z \) coordinate in the galaxy’s cylindrical coordinate system.

To determine the brightness of each bin, \textit{profile} used the mask image to specify which pixels should not be used. The remaining pixels of the bin were averaged using a detailed error model to reject any pixels more than \( 3\sigma \) away from the mean. Bins which were completely masked out were not processed. Bins with only a small number of good pixels (< 5) as well as bins in the rapidly varying central regions of the galaxy did not go through the rejection process; their pixels were simply averaged.

Since each bin had a number of pixels masked out, it was likely that the centroid of the remaining pixels was shifted somewhat from the center of the bin. The program determined the centroid of each masked bin and wrote this into a file with the calculated brightness and error.

**9.1.2 Error Modeling**

The error in our calculated brightnesses was modeled theoretically using to full advantage the well-defined noise parameters of the CCD camera. The sources of error in our data are the same as those identified by Morrison et al. (1994), and since we followed the same basic reduction procedure, the error propagation is correspondingly similar.

**Readout Noise**

For the CCD used in these observations, readout noise per pixel was 3.0 electrons, or 1.2 ADU. Since we median-combined 45 frames to obtain the final galaxy image, the noise was reduced by a factor of \( 1.22/\sqrt{45} \) to 0.22 ADU per pixel.
Photon Noise

A pixel with a brightness of $C$ ADU registered $gC$ photons when exposed to light, giving rise to photon Poisson noise of $\sqrt{gC}$, or $\sqrt{C/g}$ ADU. Once again, this noise was reduced by image combination to $1.22\sqrt{C/45g}$, or $0.115\sqrt{C}$ ADU per pixel in the final image. An average sky pixel had a value of 1877 ADU, giving rise to a typical photon noise in the sky of 5.0 ADU, or 0.2%.

Small-Scale Flatfielding Errors

Pixel-to-pixel errors were introduced into the final image when each galaxy frame was flattened by the super sky flat. These variations arise from photon noise in the individual sky flats themselves. The total error in the final galaxy frame due to this effect is given by

$$\sigma_{sff} = \frac{1}{\sqrt{gC_{sky}}} \left( \frac{1.22}{\sqrt{n_{flat}}} \right) \left( \frac{1.22}{\sqrt{n_{gal}}} \right)$$ (9.2)

where $C_{sky}$ is the mean sky background for a typical sky flat, $g$ is the gain, $n_{flat}$ is the number of combined sky flats, and $n_{gal}$ is the number of combined galaxy frames. The first factor in Eq. 9.2 is the fractional error of a single sky flat due to photon noise; the average sky value of 1877 ADU gives rise to a photon noise of 27.4 ADU, or 1.5%. The second factor comes from combining $n_{flat}$ sky flats to create the super sky flat. The third factor arises from the fact that we combined $n_{gal}$ galaxy images, each with slightly different positioning in the frame and hence independent errors.

With a sky flat photon noise of 1.5%, combining 44 sky flats and 45 galaxy frames gives us $\sigma_{sff} = 0.05\%$, a very small fractional error.

Large-Scale Flatfielding Errors

Large-scale errors in the flatfielding are a result of our inability to model the instrumental response function with complete accuracy. This effect is due mainly to the variation of the sky brightness and contamination from the wings of stars in the individual sky flats. When these flats were combined, the large-scale variations were smoothed but not completely removed. Fortunately, we had an abundance of excellent sky flats, so we were able to construct a fairly clean super sky calibration image.
The error from large-scale flatfielding effects was estimated by dividing two binned, independent super sky flats. (See Chapter 5.) The result was a 0.21% error. Since the flat fields were divided, their variances add to produce this error, so one of the independent super sky flats had a standard deviation of 0.11%. To create the final super sky flat we doubled the number of sky flats used, so this error was further reduced by a factor of $1/\sqrt{2}$ to $\sigma_{iff} = 0.07\%$.

This is again a small fractional error. However, unlike the previously considered sources of error this effect is not diminished by increasing the number of pixels in a bin. This is a limiting factor on how faint we can accurately model the galaxy light. This was the whole reason we were so careful about constructing the best possible super sky flat. It is also why half of the total open telescope time was devoted to taking calibration images rather than just galaxy frames.

Sky Background Variations

Another factor we had to take into account was the variation of the sky brightness. In Chapter 5, we saw that the sky background varied considerably from frame to frame. There was no reason to expect that the sky background remained uniform over the frame while undergoing these large fluctuations. For the purposes of this thesis, we decided it was adequate to use the large-scale flatfielding error result to determine these variations. The origin of the two effects is basically the same; one simply occurs in the sky flats and the other in the galaxy frames themselves. Thus an additional error of 0.07% was included in the model.

Surface Brightness Variations

The final source of error comes from the intrinsic surface brightness of the galaxy. Since the light we are observing comes from individual stars in the galaxy and is falling on individual pixels in the chip, it is subject to counting statistics just like any data taken in discrete units. The variance of brightness fluctuations is characterized by Tonry and Schneider (1988); for the sake of this thesis we decided to neglect the effect, which was smaller than the other error sources because of the relatively large size of the pixels in our observations (see Morrison et
al. 1994). More accurate results will be determined before publication.

Final Error Estimate

Using these values as input, profile produced an accurate error model for our data. Adding the errors in quadrature gives rise to an expression for the absolute standard deviation of a bin,

$$\sigma = \left[ \frac{C}{gn} + \frac{N_r^2}{n} + \frac{(\sigma_{sffC})^2}{n} + 2(\sigma_{ffC})^2 \right]^{1/2}$$

(9.3)

where $C$ is the value of the bin in ADU, $n$ is the number of good pixels which were averaged, and $N_r$ is the readout noise in ADU. The last term in this expression combines the contributions of large-scale flatfielding errors and intrinsic surface brightness fluctuations.

9.1.3 Perpendicular Profiles

Using the output from profile we constructed brightness profiles parallel to the minor axis of the galaxy. Only bins whose centroids fell within the central 20 pixels of each vertical strip were included. Bins with centers outside of this region were too deviant to be considered part of the profile. Figure 9.1 shows the centroids of bins around the galaxy.

Several profiles are plotted in Figure 9.2. The brightest profile corresponds to the minor axis; the brightness falls off exponentially above and below the midplane of the galaxy. In addition, the peak brightness of each profile falls off with increasing distance from the minor axis.

Is the Galaxy Warped?

To construct Figure 9.3, corresponding profiles from each quadrant of the galaxy were overlaid. This has the effect of “folding” the galaxy over the major and minor axes. The minor axis profile is a superposition of the profile above and the profile below the major axis. The other profiles are composed of four separate profiles, from the top and bottom, left and right sides of the galaxy. To get the upper ($y > 0$) and lower ($y < 0$) profiles to line up, we varied the $y$ value at which the profiles were folded until they appeared lined up by eye. This effectively defined the midplane of the galaxy. The position of these profile centers across
the galaxy indicates whether the galaxy is warped. The $y$ centers were found to vary by no more than 1 pixel over the 450 pixel length of the major axis; the resulting angle of 0.1° is within our error for the frame rotation described in Chapter 6, so this variation is quite possibly a result of that. We conclude that this galaxy does not have an appreciable stellar warp; it certainly does not have a warp of the same magnitude as that found for NGC 5907 by Morrison et al. (1994).

**Thin and Thick Disks**

Figure 9.3 is plotted on a log-linear scale, so any exponential dependence appears as a straight line. There are actually two brightness components to consider in this plot, one from the thin disk light and one from the thick disk light.

To determine the scale heights of the thin and thick disks, we fit two lines to each profile (see Figure 9.4). One line was fit to the upper points of each profile to model the brightness fall-off of the thin disk alone. A second line was fit to the lower points ($24 \leq R \leq 27$) to measure the light from the thick disk. The sum of these lines is the brightness fall-off we observe. It is worth noting that the thick disk line fits the data reasonably well down to our limiting magnitude ($\sim 29$ R mag per sq. arcsec). We conclude that either there is no luminous halo or it is extremely faint; again, this is in contrast to what Morrison et al. (1994) found for NGC 5907.

It is straightforward to find the change in magnitude corresponding to a change in distance of one scale height, $h_z$. The scale height is defined as the $z$ distance at which the ratio of the measured brightness to the peak bright $I/I_0 = 1/e$. In magnitudes this corresponds to $\Delta m = -2.5 \log[1/e] = 1.1$. Thus a change in height corresponding to a change in R magnitude of 1.1 would be equal to one scale height.

We can then calculate the scale height of each component directly from the slope of the fit line. For the minor axis profile the best fit thin disk line has a slope of $0.17 \pm 0.01$ and an intercept of $18.3 \pm 0.1$. A change in R magnitude of 1.1 produces a change in height of 6.5 pixels. For a galaxy at 9.5 Mpc, a pixel size of 2.03 arcsec yields a distance scale of 93.5 pc/pixel. The measured scale height for the thin disk is thus 600 pc.

The thick disk scale height, calculated in the same manner, turns out to be around 1.7
Figure 9.1 The true $x$ and $y$ positions of a number of bins around the galaxy, centered at $x = 1026$, $y = 1026$, are plotted. The bin centroids do not fall in a straight vertical line because they are partially masked out by the star and dust mask. Traces of bright stars can be seen in the plot, as well as a lack of data in the region of the dust lane. Also note the exponential increase in the bin height with increasing vertical distance from the galaxy.
Figure 9.2  The plot shows profiles parallel to the minor axis of the galaxy. Two profiles from the right side of the galaxy (at distances of 7.5 and 15 kpc) are shown, as well as the single profile along the minor axis, which has the highest peak. The vertical line shows the centerline of the galactic disk; the lack of data in this area is a result of the dust mask.
Figure 9.3 Three profiles are shown, overlaying data from all four quadrants of the galaxy. The distance reported at the top of each plot refers to the radial distance from the galaxy’s center, defined as $R$ in the galaxy’s cylindrical coordinate system and $x$ in frame coordinates.
Figure 9.4 The profiles from Figure 9.3 are fitted with thin and thick disk lines to determine the scale heights and thick disk normalization.
Table 9.1 Parameters of the best fit lines in Figure 9.4. The scale heights $h_z$ are in kiloparsecs, and the brightnesses at zero height, $m_0$, are in R magnitudes per square arcsecond.

The thick disk normalization is found from the magnitude difference at zero height $\Delta m_0$ of the two disk models. The intercept of the thick disk line is $21.7 \pm 0.1$, yielding $\Delta m_0 = 3.4$ R mag per sq. arcsec. This difference in magnitude corresponds to a brightness ratio of 0.04. Thus the thick disk normalization (calculated at the minor axis) is 4%.

The results for the three profiles, spanning the central 15 kpc of the galaxy, are summarized in Table 9.1. The thin disk scale height derived from the minor axis profile was used for the other two profiles. This is acceptable because there is strong evidence that the thin disk scale height is independent of the distance along the major axis (van der Kruit and Searle 1981a; Morrison et al. 1994).
Chapter 10

Conclusions

10.1 Analysis of Results

The results of the previous section improve upon previously published analysis of this galaxy (van der Kruit and Searle 1981b). The major contributions of the present work are threefold: a) the determination of the thick disk scale height and normalization, b) the demonstration that NGC 891 has no halo such as the one present in NGC 5907, and c) that the warp of the disk in NGC 891 is less than 0.1°.

Our results are also in good agreement with published research in areas of overlap. The thick and thin disk parameters obtained are similar to those measured for the Milky Way and the thin disk parameters are consistent with those determined for NGC 891 by van der Kruit and Searle (1981b). Table 10.1 shows a comparison of our results with those of van der Kruit and Searle (1981b) and with the Milky Way parameters summarized by Norris (1996).

Our estimate of the thin disk scale height is twice the accepted value for the Milky Way. This may be a difference between the galaxies or it may be due to our sole use of perpendicular profiles. The best fit line for the thin disk was only determined from five data points. In addition, there is likely to be significant contamination from thick disk light, causing an overestimation of the slope and thus the scale height. This effect appears in van der Kruit and Searle’s (1981b) results, although probably to a lesser extent.
<table>
<thead>
<tr>
<th></th>
<th>this work</th>
<th>vdKS</th>
<th>Milky Way</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thin</td>
<td>$h_z$</td>
<td>600 pc</td>
<td>500 pc</td>
</tr>
<tr>
<td>Thick</td>
<td>$h_z$</td>
<td>1.7 kpc</td>
<td>—</td>
</tr>
<tr>
<td>norm.</td>
<td>—</td>
<td>4%</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 10.1 Comparison of measured parameters for NGC 891 with those of van der Kruit and Searle (1981b, denoted vdKS) and those for the Milky Way (Norris 1996).

The thick disk parameters are close to those of the Milky Way, although these are just first estimates as well, pending a two-dimensional modeling effort.

### 10.2 Further Analysis

Future work on these data will refine the results presented in this thesis. In addition to more accurate profiles we intend to fit the galaxy with a two-dimensional model containing thin disk, thick disk, and halo components. This will allow a much more accurate estimate of the disk scale heights as well as provide a value for the total luminosity of the thick disk. We also hope to measure the radial scale length of the thick disk, a quantity which remains unknown for the Milky Way.

This two-dimensional modeling may allow the detection of (or place a more stringent upper limit on) any halo of this galaxy. This is uncharted territory for NGC 891, since van der Kruit and Searle (1981b) did not go faint enough to detect such a component. The preliminary profiles do not reflect the presence of a halo such as that detected in NGC 5907 (Morrison et al. 1994). By comparing our sensitive analysis of light far from the plane of NGC 891 with the previous analysis of NGC 5907 by Morrison et al. (1994) we hope to contribute to the understanding of dark matter (missing mass) in the universe.
Appendix A

Glossary

ADU  An ADU, or analog-to-digital unit, is the unit of measure of brightness intensity at a given image pixel. For our purposes, the intensity in ADU’s is directly proportional to the number of photons detected by a pixel in the CCD chip, with the constant of proportionality being the gain. The gain is actually specified in electrons/ADU, but a one-to-one correspondence between detected photons and registered electrons is assumed.

airmass  The amount of atmosphere through which light travels before it enters the telescope. The airmass, which determines the amount of extinction, is a function of the angle at which observations are made.

bias  The pedestal value of each pixel in a CCD chip, caused mainly by the electronics. The additive bias level must be removed before the data can be analyzed.

bias frame  A zero-integration time image, used to correct data images for bias. Several of these are normally combined to reduce the signal-to-noise ratio.

dark current  Signal caused by thermal emission of atoms in the CCD chip. It is sensitively dependent on temperature, and can be essentially reduced to zero by cooling the chip. At room temperature, the dark current is strong enough to saturate the chip within just a few seconds.

dark frame  An image taken with the camera shutter closed, usually for the same integration time as the object images. It is used to remove the additive dark current from all images of significant exposure time.

dark matter  The “missing mass” of the universe. It has been observed to exist from rotation curves of galaxies, but no luminous counterpart has been yet found. There are many theories about the nature of dark matter, from exotic subatomic particles to large star-like objects called brown dwarfs.

detection threshold  The daophot parameter that determines the minimum peak above background for an object to be identified as a star. The detection threshold is quoted either in counts (ADU) or in multiples of sigma, the standard deviation of the noise.
dithering  The technique of slewing the telescope slightly between successive exposures of the same field. This ensures that an object does not fall on the same part of the chip in multiple exposures, thereby minimizing the effects of bad pixels and other sensitivity variations.

edge-on  Refers to a spiral galaxy which has its axis of rotation perpendicular to our line of sight. This enables us to observe the brightness above and below the galactic plane without interference from the dust or stars in the spiral arms.

extinction Reduction of signal caused by the earth’s atmosphere. It is dependent on the airmass through which the observations are being made, as well as the filter being used. The extinction in the R-band for Kitt Peak is 0.1, meaning a star observed at 1 airmass would have a measured magnitude 0.1 fainter than its true magnitude.

fitting radius  The radius (in pixels) to which the analytic component of a point spread function is fit to a star.

flat field An image of a uniform light source, used to correct data images for systematic brightness variations. To get the most accurate response, it is important to use a light source which mimics the object source in color.

look-up table  The empirical component of the point spread function. The look-up table records residuals from the analytic fit, and uses these to more accurately model the stellar profile.

magnitude  The measurement scale of stellar brightness. A difference of 5 magnitudes is defined to equal a factor of 100 in brightness, with higher magnitudes denoting fainter stars. Thus the magnitude scale is approximately given by

\[ m_1 - m_2 = -2.5 \log \frac{I_1}{I_2} \]

where \( m \) is magnitude and \( I \) is intensity of a star. The zero point for this scale is defined by the star Vega. A useful fact is that a difference in magnitude of 0.01 corresponds to a factor in brightness of 1%.

metallicity  The metal content of a star, often given in terms of the solar metal abundance. “Metal” refers to any element heavier than helium.

NGC  Stands for New General Catalogue of Nebulae and Clusters of Stars, which is a listing of various celestial objects compiled by John Dreyer in 1888. NGC 891 is the 891st entry in this catalog.

overscan The region of an image frame which is not exposed to light; in our data, it was columns 2049 to 2080. It is used to determine the average bias level of the chip for a particular image.

parsec (pc)  Astronomical unit of length. A parsec is defined as the distance at which the radius of the Earth’s orbit (1 astronomical unit or AU) subtends an angle of 1 arcsecond. 1 pc = 3.26 ly = 3.06 \times 10^{16} m

photometric  Photometric conditions are conditions under which stellar brightnesses can be accurately measured. Any feature which filters light in a non-uniform manner, such as clouds, would cause non-photometric conditions in which stars would be seen to brighten and dim throughout the night. Note that additive effects such as background variation and light pollution would not destroy photometric conditions; they would merely lower the signal-to-noise ratio of the image.
**Photometry**  Measurement of the brightness of an astronomical object, usually a star. There are two basic types of stellar photometry, aperture and PSF. Aperture photometry is treated in Appendix B, and PSF photometry is treated in Chapter 8, where it is used to create the star mask for the final galaxy image. This entire work focuses on another form of photometry, “surface” photometry.

**Point spread function**  The function which models the stellar profile. Our PSF was composed of an analytic function and empirical look-up table; the combination of the two produced a more accurate model profile than either one alone. It is used mostly to perform photometry in crowded fields, although we also used it to determine the size of the stellar masks in Chapter 8.

**Population I**  Young, metal-rich stars, found in the star-forming regions of the spiral arms. These stars have condensed from the remnants of supernovae, which include heavy elements up to iron. They cover the whole range of spectral types, from hot, massive stars to cool, faint stars. Our Sun is an example of an old Population I star.

**Population II**  Old, red, metal-weak stars, found in the bulge and halo of a spiral galaxy. These stars condensed long ago before many supernovae had produced heavy metals and spread them through interstellar space. Since that time, the hotter, more luminous stars of this population have burned themselves out, leaving only the faint red dwarfs.

**PSF**  See point spread function.

**PSF radius**  The radius of the point spread function (in pixels). When PSF photometry is performed, the PSF is fit to each star out this radius. We used a PSF radius of 15 pixels.

**PSF stars**  Several bright, isolated stars from which an accurate point spread function can be calculated by an iterative fitting procedure.

**Radiation events**  Signal in CCD images caused by things such as cosmic rays. These are removed by combining several images and rejecting a fraction of the highest pixels.

**Readout noise**  Noise arising from the amplifiers and other electronics which read the data from the chip and convert it into ADU. It is well-characterized for a given chip; for the one used in our observations, the readout noise was 3.0 electrons.

**Red dwarf stars**  Faint, low-mass stars which make up a large portion of Population II.

**Register**  Images which are lined up vertically, as measured by $x$ and $y$ coordinates or by star location, are said to be registered.

**Saturation**  The point at which data from a CCD chip loses its linearity and becomes unreliable. In most cases, this is caused by limitations in the conversion from electrons to ADU, not by limitations of the chip itself.

**Scale height**  Vertical distance from the plane of a galaxy at which the brightness, and thus the mass density, falls to $1/e$ times its highest value. In this text, scale height is written as $h_z$.

**Scale length**  Radial distance from the center of a galaxy at which the brightness, and thus the mass density, falls to $1/e$ times its highest value. In this text, scale length is written as $h_R$. 

69
seeing  A parameter describing the quality of observations based on the average resolution that can be obtained. Note that conditions can be perfectly clear but still provide bad seeing if there is significant turbulence in the air. Seeing is often defined in terms of the full width at half maximum of a typical stellar profile.

sky flat  A type of flat field in which a region fairly free of stars is imaged. The resulting calibration image possesses a color and brightness variation similar to the object frames, but suffers from low signal-to-noise and contamination due to stars.

solar mass  The mass of the sun, designated as $M_{\odot} = 2.0 \times 10^{30}$ kg.

standard stars  Stars whose magnitudes are accurately known, used for calibrating results to the standard research. Many of the standard stars in use today were observed by Landolt (1983, 1992).

thermal noise  The $\sqrt{N}$ noise associated with $N$ counts from the dark current. Thermal noise is often the name mistakenly given to dark current. Dark current is a real signal; thermal noise is the noise of that signal.

wings  (of a star) The outer regions of a star. These regions contain a small amount of light compared to the inner region, but cause a significant variation in the measured sky background.
Appendix B

Aperture Photometry

Aperture photometry is used to find the brightness of relatively bright, isolated stars, such as the test stars from Chapter 5 and the standard stars from Chapter 7. The program used for all the stellar photometry performed in this project was daophot, originally written by Stetson (1987) and later included in IRAF as a separate package.

All digital photometry consists of two steps: first identifying the stars and then performing the actual brightness measurements. To locate the stars, a task called daofind is used. Input to the task includes the full width at half maximum (FWHM) of a typical stellar profile and the typical peak brightness of a star above the background, or the detection threshold. daofind then searches through each specified image for features which match this FWHM and have a peak above the detection threshold. The coordinates of all located stars are output to a file.

To begin the second step, this coordinate file is input to a second task called phot. This task performs the actual aperture photometry. The flux through an aperture centered on a star is measured. The aperture, whose radius is specified as input to the task, must be large enough to collect the majority of light from the star and not be affected by seeing. On the other hand, it must be small enough that it does not include any bad pixels or any light from other stars or radiation events. For most photometry, an aperture size of 4 to 5 times the FWHM is normally used (Massey and Davis 1992); for our data, the FWHM was about...
2.6 pixels, so the aperture radius would be 10-15 pixels.

The next step is to account for the sky brightness. The intensity of each pixel in a thin annulus centered on the star is measured. The sky brightness can be then determined from these values in a number of ways. For our purposes, it was best to use the mode of these pixels. **phot** performs an iterative procedure of calculating the mode, then rejecting any pixels higher or lower than a specified multiple of the standard deviation. The iterations continue until there are no more rejections or the specified maximum number of iterations is reached. The final value is taken to be the sky value for the given star.

Figure B.1 shows a typical stellar profile and the regions that would be measured to find the stellar flux and sky brightness. The inner shaded region represents the aperture, while the outer shaded region represents the sky annulus. The inner radius and width of the sky annulus are given as input to **phot**. It is important to keep the inner radius relatively large (several times the FWHM) so that light from the wings of the star is minimized. The width of the annulus must be large enough to contain a reasonable number of sky pixels, but it must not be too large that it falls on another star or several bad pixels. While the mode is fairly robust under these conditions, the standard deviation is not, and hence the rejection procedure could be adversely affected.

Once the sky value is obtained, it is multiplied by the number of pixels in the stellar aperture and subtracted from the flux measured through that aperture. This is equivalent to subtracting the sky value from every pixel in the aperture. Thus the additive sky background is removed and only the light due to the star is left.

The flux is changed into a magnitude by definition of an arbitrary (but consistent) zero-point magnitude within **IRAF**. For these reported magnitudes to have any external meaning, they must be calibrated to standard magnitudes (see Chapter 7). This calibration scales the flux by a multiplicative factor; for the logarithmic magnitude scale, of course, this corresponds to an additive zero-point offset.
Figure B.1 A profile of a typical, isolated star. The inner shaded region is the aperture; light from this region is summed to determine the total flux of the star. The outer region is the sky annulus, from which the level of the sky background is measured.
Appendix C

Program Listings

This appendix contains complete listings of all programs (external to IRAF) used in this thesis. All programs were written in Fortran by Heather Morrison. Modifications for the NGC 891 data, made by the author, are noted in the code.

C.1 newmode

```
MODE -- calculates the mode for a CCD scan
version for real data

ismooth is the amount the histogram gets smoothed before its
maximum is found; percent tells you which percentile to
go down to when finding trimmed mean from smoothed histogram

history:
march 18 mods by ph for 1024 image

when we find the centroid of the histogram after smoothing
we can introduce noise due to the fact that bins do not get included
symetrically. This is the noise we see if we use a high percentile
like .5 or .25. It gets better at .10 as the change in the number
of counts -- one bin is small.

ph oct93 Adds in fractional portion of last bin on either side
of histogram above
the percentage cut -- make a big improvement to the flats.
feb95 hlm mods to use for estimating sky level for imcombine
since IRAF wont do it properly
```

-----------------------------
program mode
character*50 image, errmsg, outfile
integer ismooth, hista(10000), max, pointer
integer imin, imin2, imin3, ier, axlen(7), naxis, dtype
real row(2048), est, percent
real sum
percent = 0.9
ismooth = 7

c
--- Read name of file containing image names from STDIN
print *,'Enter name of input file: ', image
read (*,*), image
open(1,file=image,status='old')
print *, 'Was able to open the input file.'

c
--- Read name of output file from STDIN
print *,'Enter name of output file: ', outfile
read(*,*), outfile
open(2,file=outfile,status='unknown')
print *, 'Was able to open the output file.'

c
--- loop thru input images
do ii=1,100
   read(1,'(a50),end=99) image
   if (image(1:3).eq."---") then
      print *,'
      print *, '--------------------',
      print *, 'Running set 2...
      print *, '--------------------',
      goto 90
   endif
   print *, 'Reading image ',image

c
--- Open the image and work out its size.
call imopen (image, 1, imin, ier)
if (ier.ne. 0) then
   print *, 'problem opening input image'
goto 91
endif
call imgsiz (imin, axlen, naxis, dtype, ier)
if (ier.ne. 0) goto 91

c
--- zero the histogram array
do 15 i=1,10000
   hista(i) = 0
15 continue

c
--- histogram the columns into separate bits of hista.
do 20 i=1, axlen(2)

c
--- Extract a line from image
call imgl2r (imin, row, i, ier)
if (ier.ne. 0) goto 91
--- Add to histogram array
  do 30 j=1,2048
    if(row(j).ge.0.and.row(j).lt.10000) then
      k = int(row(j))
      hista(k) = hista(k) + 1
    endif
  enddo
  continue
  continue

--- Work out the 'trimmed mean' of histogram using estsky.
call estsky (hista, issmooth, percent, 1, est)

--- Write the sky estimate out.
write(2,*) est

c --- Close the input image and quit.
call imclos (imin, ier)
if (ier .ne. 0) goto 91
end do

99 close(1)
close(2)
stop

--- Error handler.
call imemsg (ier, errmsg)
write (*, '(Error: ', a80)') errmsg
stop

end subroutine estsky (eta, issmooth, percent, cll, est)

subroutine to estimate a sky value
c the eta function (histogram of data values) is first smoothed
c by triangular smoothing over isMOOTH - 2 points. the maximum
c in the smoothed eta function is then set to the modal sky value.
c the large 1-D histogram file eta is fed to the subroutine, and
c cll tells it where the histogram of interest starts.
c spaghetti fortran variable names from KUR

c the arguments of the subroutine call are
  input :  eta  =  histogram of intensity values
           isMOOTH = no. of points in triangular smooth + 2
           percent = percentile points to sum histogram over to
c           get location estimate
           cll  =  start point for this histogram
        output:  est  =  best estimate of location

c integer    cll,clu
integer    eta(10000), imax
real        ccc(10000)

clu = 10000

c smooth eta and store result in ccc. look for maximum also.
n=(ismooth-3)/2
mine=max(0,cl1+n)
maxe=min(10000,clu-n)
n1=n+1
sum=float(n1)*eta(mine)
do 10 i=1,n
   il=n1-i
   sum=sum+float(i)*(eta(mine-il)+eta(mine+i))
10 continue
delta=1.0/float(n1*n1)
smeta=sum*delta
amaxeta=0.0
do 15 i=mine,maxe-1
   k = i-cl1+1
   ccc(k)=smeta
   if(smeta.gt.amaxeta) then
      amaxeta=smeta
      imax=k
   endif
   do 20 j=1,n1
      sum=sum-eta(i-j+1)
      sum=sum+eta(i+j)
20 continue
smeta=sum*delta
15 continue
k = i-cl1+1
ccc(k) = smeta

fill in the bits of ccc which are missing cos you cant smooth there

do 25 i=1,n
   ccc(i) = eta(cl1+i-1)
   ccc(10000-i+1) = eta(clu-i+1)
25 continue

calculate measure of location
c
call trimm (ccc, imax, amaxeta, percent, est)
c
iterate around again; trimm has updated imax and amaxeta
call trimm (ccc, imax, amaxeta, percent, est)
c
compensate for truncation in initial histogramming (0-0.999 goes to 0
c which is bin 1) by subtracting 0.5 from result
est = est - 0.5
return
c
subroutine trimm (ccc, imax, amaxeta, percent, est)

give it the smoothed histogram ccc, its maximum
c
and the percentile you want to go down to, and it will calculate
a weighted mean of all the histogram values (weighted by no.
of points in that bin) down to the bin with (say) 10% of the maximum
counts.

input:      ccc     smoothed histogram
imax       bin number of maximum
amaxeta    maximum
percent     percentile to sum to
returned:   est     estimate of location from weighted sum
imax       bin no corresponding to est
amaxeta    value in that bin

work down from peak of histogram in both directions
limit calc to max of 50 bins either side, usually will hit
percent cut limit first.

ph dec 93 add in the value in the bin above the cut only
so dont get noise due to bin discreteness in mean
this should be the final correct answer.

real          ccc(10000), amaxeta, cut, weight, msum, est, percent
integer       imax, i
real          binfrac

center bin
cut = amaxeta * percent
weight = amaxeta - cut
msum = (amaxeta - cut) * float(imax)

lower bins
do 10 i=1, min0(50,imax-1)
   if(ccc(imax-i).ge.cut) then
      binfrac = ccc(imax-i) - cut
      msum = msum + binfrac*float(imax-i)
      weight = weight + binfrac
   else
      goto 20
   end if
10 continue

upper bins
20 do 30 i=1, min0(50,10000-imax)
   if(ccc(imax+i).ge.cut) then
      binfrac = ccc(imax+i) - cut
      msum = msum + binfrac*float(imax+i)
      weight = weight + binfrac
   else
      goto 40
   end if
30 continue

40 est = msum / weight
imax = nint(est)
amaxeta = ccc(imax)
return
end
C.2 spike mode

c MODE -- calculates the mode for a CCD scan
version for real data

ismooth is the amount the histogram gets smoothed before its
maximum is found; percent tells you which percentile to
go down to when finding trimmed mean from smoothed histogram

history:
march 18 mods by ph for 1024 image

when we find the centroid of the histogram after smoothing
we can introduce noise due to the fact that bins do not get included
symetrically. This is the noise we see if we use a high percentile
like .5 or .25. It gets better at .10 as the change in the number
of counts -- one bin is small.

ph oct93 Adds in fractional portion of last bin on either side
of histogram above
the percentage cut -- make a big improvement to the flats.

feb95 hlm mods to use for estimating sky level for imcombine
since IRAF wont do it properly

----------------------------------------------

program mode

character*50           infile, errmsg, outfile, image
character*9            rawfile, topfile
character*12           smoothfile
character*2            number(50)

integer               ismooth, max, ii
integer               inmin, inmin2, inmin3, ier, axlen(7), naxis, dtype
integer               lowlim, uplim
real                  hista(10000), row(2048), est, percent
real                  sum

percent = 0.9
ismooth = 13
max = 0
lowlim = 1
uplim = 10000

rawfile = "hist.raw."
smoothfile = "hist.smooth."
topfile = "hist.top."

*--- Read name of file containing image names from STDIN
print *, 'Enter name of input file: ',
read (*,*) infile
open(1, file=infile, status='old')
print *, 'Was able to open the input file.'

*--- Read name of ouput file from STDIN
print *, 'Enter name of output file: '
read(*,*) outfile
open(2, file=outfile, status='unknown')
print *, 'Was able to open the output file.'

c    --- Initialize stupid number array
call numinit(number)

c    --- Print out some prelim info
print *, ''
print *, 'Width of median smoothing window is ', ismooth, '.'

c    --- loop thru input images
do ii=1,100
   read(1,'(a50)', end=99) image
   if (image(1:3).eq.'---') then
      print *,''
      print *, '-----------------------------'
      print *, 'Running set 2...
      print *, '-----------------------------'
goto 90
   endif
   print *, '',
   print *, 'Reading image ', image
   c    --- set pixel directory
call imsdir ('HDR$pics'/)

c    --- Open the image and work out its size.
call iomopen (image, 1, imin, ier)
   if (ier .ne. 0) then
      print *, 'problem opening input image'
goto 91
   endif
   call imgsiz (imin, axlen, naxis, dtype, ier)
   if (ier .ne. 0) goto 91

c    --- zero the histogram array
do 15 i=1,10000
   hista(i) = 0.0
15 continue

c    --- histogram the columns into separate bits of hista.
do 20 i=1, axlen(2)

c    --- Extract a line from image
call imgl2r (imin, row, i, ier)
   if (ier .ne. 0) goto 91

c    --- Add to histogram array
do 30 j=1,2048
   if (row(j).ge.0.and.row(j).lt.10000) then
      k = int(row(j))
      hista(k) = hista(k) + 1.0
   endif
   if (row(j).gt.maxval) maxval=row(j)
   if (row(j).lt.minval) minval=row(j)
30 continue
continue

-- Write the raw histogram to file and find maximum
open (20, file="hist.raw."//number(ii), status="unknown")
print *, 'Writing raw histogram file " hist.raw." // number(ii) //',
do 40 i=lowlim, upplim
   write (20,*) i, hista(i)
   if (hista(i).gt.histax(max)) max=i
40 continue
print *, 'Max raw number is', histax(max), ' at bin ', max
close (20)

--- Work out the 'trimmed mean' of histogram using estsky.
call estsky (hista, ismooth, percent, lowlim, upplim, &
   max, number(ii), est)

--- Write the sky estimate out.
   write(2,*) est
   print *, 'The mode is ', est

--- Close the input image and quit.
call imclos (imin, ier)
   if (ier.ne. 0) goto 91
   print *, 'Done image ', image
90 end do
99 close(1)
close(2)
stop

--- Error handler.
call imemsg (ier, errmsg)
write (*, '('' Error: '', a80')') errmsg
stop
end

subroutine estsky (ccc, ismooth, percent, lowlim, upplim, max, &
   number, est)

subroutine to estimate a sky value
the hista function (histogram of data values) is first smoothed
by triangular smoothing over ismooth - 2 points. the maximum
in the smoothed hista function is then set to the modal sky value.
the large 1-D histogram file hista is fed to the subroutine, and
lowlim tells it where the histogram of interest starts.
spaghetti fortran variable names from KUR

the arguments of the subroutine call are
   input : hista = histogram of intensity values
   ismooth= no. of points in median smooth + 2
   percent= percentile points to sum histogram over to
   get location estimate

c  lowlim   = start point for this histogram
max      = maximum
output:  est   = best estimate of location

character*2    number
integer   lowlim, upplim, max
real      ccc(100000), temp(20)

c   -- Perform a median smooth
n=(isMOOTH-1)/2
do 10 i=lowlim+n,upplim-n
   20 do 20 j=-n,n
      temp(j+n+1)=ccc(i+j)
   continue
   call sort(issmooth,temp)
   ccc(i)=temp(n+1)
10 continue

c   -- fill in the bits of ccc which are missing cos you can't smooth there
   do 25 i=1,n
      ccc(i) = ccc(lowlim+i-1)
   continue

   temp(upplim-i+1) = ccc(upplim-i+1)
25 continue

c   -- Write the smoothed histogram to file and find maximum
max = lowlim
open (21, file="hist.smooth."/number, status="unknown")
p0int *, 'Writing smoothed histogram file ' 'hist.smooth.'/number//'','
do 30 i=lowlim,upplim
   continue
   print *, 'Max smoothed number is',ccc(max),' at bin ',max
30 close (21)

c   -- calculate measure of location
   call trimm (ccc, max, ccc(max), percent, number, est)

c   -- iterate around again; trimm has updated imax and amaxeta
   call trimm (ccc, max, ccc(max), percent, number, est)

c   compensate for truncation in initial histogramming (0-0.999 goes to 0
which is bin 1) by subtracting 0.5 from result
est = est - 0.5
return
end

subroutine trimm (ccc, imax, amaxeta, percent, number, est)
c give it the smoothed histogram ccc, its maximum
and the percentile you want to go down to, and it will calculate
a weighted mean of all the histogram values (weighted by no.
c of points in that bin) down to the bin with (say) 10% of the maximum
c counts.
input: ccc smoothed histogram
imax bin number of maximum
amaxeta maximum
percent percentile to sum to
returned: est estimate of location from weighted sum
imax bin no corresponding to est
amaxeta value in that bin

work down from peak of histogram in both directions
limit calc to max of 50 bins either side, usually will hit
percent cut limit first.

ph dec 93 add in the value in the bin above the cut only
so dont get noise due to bin discreetness in mean
this should be the final correct answer.

character*2 number
real ccc(10000), amaxeta, cut, weight, msun, est
integer imax, i
real binfrac, percent

center bin
cut = amaxeta * percent
weight = amaxeta - cut
msun = (amaxeta - cut) * float(imax)

open (23, file="hist.top."/number, status="unknown")
print *,'Writing top of histogram file ' 'hist.top.'/number//''

lower bins
   do 10 i=1, imax-1
      if(ccc(imax-i).ge.cut) then
         binfrac = ccc(imax-i) - cut
         msun = msun + binfrac*float(imax-i)
         weight = weight + binfrac
         write (23,*) imax-i, ccc(imax-i)
      else
         goto 20
      end if
10 continue

upper bins
   do 20 i=1, 10000-imax
      if(ccc(imax+i).ge.cut) then
         binfrac = ccc(imax+i) - cut
         msun = msun + binfrac*float(imax+i)
         weight = weight + binfrac
         write (23,*) imax+i, ccc(imax+i)
      else
         goto 40
      end if
20 continue

close (23)

40 est = msun / weight
imax = nint(est)
amaxeta = ccc(imax)
return

end
C.3 radial

c make a psf using daophot; produce an image of it using seepsf;
c use pradprof to produce an incoherent sort of radial profile;
c use Unix sort to sort in radius order; then use this program to
c average numbers in radial bins (with trimming) and write out
c a radial profile.

imPLICIT NONE

REAL R(10000), TOP
REAL X(1000), XX(10000), SIG, TRIM, MEAN, REALTRIM, ORDER(1000)
REAL TRIMMED(1000), RAD
INTEGER I,J,N,NTRIM, NUM, K

OPEN(1, FILE = 'PRADPROF.OUT', STATUS = 'OLD')
OPEN(2, FILE = 'RAD.SHORT', STATUS = 'NEW')

TRIM = 0.1

c read pixel intensities into arrays XX and R (for radius)

do i = 1, 10000
  READ(1, *, END = 99) R(I), XX(I)
end do

99

NUM = I - 1

c sort into bins of similar radius

do i = 1, 16
  TOP = FLOAT(I) - 0.5
  RAD = 0
  K = 0
  DO J = 1, NUM
    IF (R(J) .LT. TOP .AND. R(J) .GE. TOP - 1) THEN
      K = K + 1
      X(K) = XX(J)
    END IF
  END DO

  N = K

c If there are less than 10 values, just average them, otherwise trim sig

IF (N .EQ. 1) THEN
  MEAN = X(1)
  IF (MEAN .GT. 0) THEN
    WRITE(2, 10) TOP - 0.5, MEAN, ALOG10(MEAN)
  ELSE
    WRITE(2, 10) TOP - 0.5, MEAN
  END IF
ELSE IF (N .LT. 10) THEN
  CALL STDEV(X, N, MEAN)
  IF (MEAN .GT. 0) THEN
    WRITE(2, 10) TOP - 0.5, MEAN, ALOG10(MEAN)
  ELSE

85
write(2,10) top=0.5, mean
end if

10 else
    call sort(n,x,order)
    realtrim = float(n)*trim
    ntrim = jint(realtrim) ! nearest integer
    if(ntrim.eq.0) ntrim=1
    do j=1,n-2*ntrim
        trimmed(j) = order(j+ntrim)
    end do
    call stdev(trimmed,n-2*ntrim,mean,sig)
    if(mean.gt.0) then
        write(2,10) top=0.5,mean,alog10(mean)
    else
        write(2,10) top=0.5, mean
    end if
end if
end do

close(1)
close(2)
end

subroutine stdev(x,n,mean)
    real x(10000), sig, mean
    real sumx,sumxsq
    integer n

    sumx = 0.
    do i=1,n
        sumx = sumx + x(i)
    end do
    mean = sumx/float(n)
    return
end

Subroutine sort(n,inarray,outarray)
* from Blair Phillips, much better than Kavan's
    implicit none
    integer n
    real inarray(n),outarray(n)
    integer i,sorted,ismall
    real small
    do i = 1,n
outarray(i) = inarray(i)
enddo

* outarray(1..sorted-1) is sorted
*

dosorted = 1,n-1
ismall = sorted
small = outarray(ismall)
doi = sorted+1,n
if (outarray(i).lt.small) then
  ismall = i
  small = outarray(ismall)
endif
enddo
outarray(ismall) = outarray(sorted)
outarray(sorted) = small
enddo
return
end
C.4  mask

```fortran
program makemask

implicit none

integer       ncols, nlines
real          psfmag, limit
parameter (ncols=2045, nlines=2048, psfmag=12.514)
parameter (limit=1.0)

case*80      image, errmsg, dustfile, radfile, magfile
integer       im, im2, ier, axlen(7), naxis, dtype
integer       i, j, k, ix, ixc, iyc, id
integer       xstart, xend, ystart, yend, iieer
real          buf(ncols), mask(ncols,nlines), dust(ncols,nlines)
real          radius(34), counts(34), mag, delcounts
real          exclrad(300), xc, yc, compmag, rad, xrad
real          line1, line2, line3, line4

c   --- Get image name for mask image.
   write (*, '(" enter mask image name: ",$)')
   read (*, '(a80)') image

c   --- Get pixel datatype (real)
   dtype = 6

c   --- Set pixel directory to HDR$pix/.
   call imkdir ('HDR$pix')

c   --- Image size:
   axlen(1) = ncols
   axlen(2) = nlines
   naxis   = 2

c   --- Create the image.
   call imcrea (image, axlen, naxis, dtype, ier)
   if (ier .ne. 0) goto 91

c   --- Open the image for writing.
   call imopen (image, 3, im, ier)
```

88
if (ier .ne. 0) goto 91

c    --- Set pixel directory to HDR$pix/.
c    call imsdirl ('HDR$pix/')

c    --- prompt for name of dust mask image.
c    write (*,'(Enter existing dust mask image name: '',$)')
c    read (*,'(a80)') image

c    --- Open the image for reading only.
c    call imopen (image, 1, im2, ier)
    if (ier .ne. 0) goto 91
    c    --- Read image into dustmask array.
do i=1,nlines
        call imgl2r(im2,buf,i,ier)
        if (ier .ne. 0) goto 91
        do j=1,ncols
            dust(j,i) = buf(j)
        end do
    end do

c    --- Open star exclusion radius file (psf2.smooth for 5907 I).
c    --- (rad.short for 891 R)
c    write (*,'(Exclusion radius filename: '',$)')
c    read (*,'(a80)') radfile
    open(2, file=radfile, status='old')

c    --- Open file with position and magnitude of each star
    c    --- (ngc891.als.25 for 891 R).
c    write (*,'(Allstar output filename: '',$)')
c    read (*,'(a80)') magfile
    open(3, file=magfile, status='old')

c    --- read header from allstar output file
    do i=1,44
        read(3,*)
    end do

c    --- Initialise mask array by copying dustmask over
    do i=1,ncols
        do j=1,nlines
            mask(i,j) = dust(i,j)
        end do
    end do

c    --- STARS:

c    --- Work out what radius a star of a given magnitude goes out to.
c    c    --- Magnitudes from 10 to 40; psf star has magnitude psfmag (12.514);
c    c    --- hlm(note that the limit is now 0.5 counts since 1 count is I=27.6)
c    c    --- limit is defined by parameter limit
    open (20, file='magfile.out', status='unknown')
do i=1,16
    read(2,*) radius(i), counts(i)
end do

do i=1,300
    mag = 9.9 + float(i)/10.
    delcounts = 10**((psfmag-mag)/2.5)
    do j=1,16
        if(counts(j)*delcounts .lt. limit) goto 20
    end do
    exclrad(i) = radius(j)
    write(20,*) mag, exclrad(i)
end do
close (20)

c --- Read in the position and magnitude of each star.
do i=1,100000
    write(4,*) i
    read(3,*,end=40) id, xc, yc, mag
read(3,*)
    if(dust(anint(xc),anint(yc)).eq.0) goto 90
    if(mag.lt.10) then
        rad = 30.
    else
        do j=1,300
            compmag = 9.9 + float(j)/10.
            if(mag.lt.compmag) then
                rad = exclrad(j)
            go to 30
        end if
end do
end if

30    ixc = anint(xc)
    iyc = anint(yc)

    c --- Write zeroes into the appropriate portion of the mask array.
    c --- Center row:
do k=1,anint(2*rad)
    ix = ixc - anint(rad) -1 + k
    if(ix.ge.1.and.ix.le.ncols.and.iyc.ge.1
        .and.iyc.le.nlines) then
        mask(ix,iyc) = 0.
    end if
end do
    c --- Rest:
    if(rad.gt.1) then
        do j=1,anint(rad)
            xrad = anint(sqrt(rad*rad-float(j)**2))
            do k=1,xrad*2
                ix = ixc-xrad-1+k
                if(ix.ge.1.and.ix.le.ncols) then
                    if(iyc+j.ge.1.and.iyc+j.le.nlines) then
                        mask(ix,iyc+j) = 0.
                    end if
                    if(iyc-j.ge.1.and.iyc-j.le.nlines) then
                        mask(ix,iyc-j) = 0.
                    end if
                end if
            end do
        end do
    end if
end if
end do
end do
90 end do
40 continue

c --- EDGES:
c --- Because we had to rotate the image to make the galaxy parallel
c --- to the x axis, there are bits of the image that should go.
do i=1,ncols
do j=1,nlines
    line1 = 589.126-(float(i))*2.15441
    line2 = -543.459+(float(i))*0.455625
    line3 = 5209.96-(float(i))*2.08867
    line4 = 1632.1+(float(i))*0.464297
    c line1 = 405 - (float(i))*0.426
    c line2 = 2.36*(float(i))-1566
    c line3 = 1329 + 2.36*(float(i))
    c line4 = 2048 - (0.426 * (float(i)-1013))
    if (j.le.aint(line1)) then
        mask(i,j) = 0
    else if (j.le.aint(line2)) then
        mask(i,j) = 0
    else if (j.ge.aint(line3)) then
        mask(i,j) = 0
    else if (j.ge.aint(line4)) then
        mask(i,j) = 0
    end if
end do
end do
c --- Write the mask data into the image, line by line.
do j=1,nlines
    do i=1,ncols
        buf(i) = mask(i,j)
    end do
    call impl2r(im, buf, j, ier)
end do
c --- Close the image and quit.
close(2)
close(3)
call imclos (im, ier)
call imclos (im2, ier)
if (ier.ne. 0) goto 91
stop
c --- Error exit.
91 call imemsg (ier, errmsg)
write (*,('('' Error: '', a80)')) errmsg
stop
end
C.5 profile

c --- Bins up a galaxy image into vertical cuts for profiles parallel
c --- to minor axis
c --- first hack at 4565 data
c --- EDM 4/6/96 updated for 891 data

    program profile

    implicit none

    integer cols, lines, firstwid, xc, yc
    real epdn

c --- EDM new parameters for 891
    parameter(cols=2045,lines=2048,firstwid=3,xc=1026,yc=1025)
    parameter(epdn=2.5)

    character*80 image, errmsg, estfile

    integer ncols, nlines
    integer im, im2, ier, axlen(7), naxis, pixtype
    integer hr, hz, ycen, ywid, ywidold
    integer k, ystart, yend, ycenold
    integer ieeer, left(2), ny
    integer myhandler, ieee_handler
    integer mask(cols,100)

    real work(cols,100), grid(cols,lines)

    c --- Get image name for image to be binned up.
    write (*, '(' enter image name: '"',$')
    read (*,'(a80)') image

    c --- Open the image and find out how big it is, etc.
    call imopen (image, 1, im, ier)
    if (ier .ne. 0) goto 91
    call imgsiz (im, axlen, naxis, pixtype, ier)
    if (ier .ne. 0) goto 91
    ncols = axlen(1)
    nlines = axlen(2)

    c --- Get image name for mask.
    write (*, '(' enter mask name: '"',$')
    read (*,'(a80)') image

    c --- Open the mask image.
    call imopen (image, 1, im2, ier)
    if (ier .ne. 0) goto 91
c --- Open file for vertical profile data.
    write(*,'("Output filename: ",$')
read(*,'(a80)') estfile
open(1,fil=estfile,status='new')

--- Prompt for scale height for grid pattern construction.
write(*,'(" enter hz in pixels: ",$')
read(*,*) hz

--- I used 30 for the R band paper; 11 for the I band (Schmidt) data
--- EDM used 11 here for 891
    ywid = firstwid
    ycen = yc

--- CENTER ROWS (Y):
c --- Read first 3 rows of image and mask into working arrays work and mask.
call arrays(ycen,ywid,work,mask,ncols,im,im2)

--- Split this slice up into x-bricks, average and write out.
call brick(ncols,nlines,hr,xc,ycen,ywid,
    work,mask,grid,firstwid,epdn)

--- OTHER ROWS (Y):
do k=1,2
    ywidold = ywid
    ycenold = ycen
20    call widcalc(ycen,ywidold,ycenold,hz,k,ywid,ycen)
        if(ywid.eq.99.and.ywidold.ne.99) then
            left(k) = ycenold + (-1)**k*ywidold/2
        end if

--- Check whether we have got to the top or bottom of the image.
ystart = ycen - ywid/2
yend = ystart + ywid - 1
if(ystart.ge.1.and.yend.le.nlines) then

--- Write new rows into working arrays.
call arrays(ycen,ywid,work,mask,ncols,im,im2)

--- Divide it up into x-bricks;
call brick(ncols,nlines,hr,xc,ycen,ywid,
    work,mask,grid,firstwid,epdn)

--- Unless we are at the top or bottom of the image, go back for
--- another strip.
    if(ystart.ne.1.and.yend.ne.nlines) then
        go to 20
    else
        ycen = yc
        ywid = firstwid
    end if
else if(yend.gt.nlines) then
    yend = nlines
    ywid = 2*(yend-ystart)/2+1
    ycen = ystart + ywid/2

  c --- Write new rows into working arrays.
  call arrays(ycen,ywid,work,mask,ncols,im,im2)

  c --- Divide it up into x-bricks;
  call brick(ncols,nlines,hr,xc,ycen,ywid,
             work,mask,grid,99,epdn)

  c --- Reset this (for the halibut).
  ycen = yc
  ywid = firstwid
  else
      ystart = 1
      ywid = 2*(yend-1)/2+1
      ycen = ystart + ywid/2
  end if

  c --- Write new rows into working arrays.
  call arrays(ycen,ywid,work,mask,ncols,im,im2)

  c --- Divide it up into x-bricks;
  call brick(ncols,nlines,hr,xc,ycen,ywid,
             work,mask,grid,99,epdn)

  c --- Reset for the top half.
  ycen = yc
  ywid = firstwid
end if
end do

  c --- DONE!   AAAAAARRGH!

c  --- Close the image and quit.
  close(1)
  call icmclose (im, ier)
  call icmclose (im2, ier)
  if (ier .ne. 0) goto 91

  stop

  c  --- Error exit.
  91  call icmsg (ier, errmsg)
      write (*, '('' Error: '''', a80)') errmsg

      stop
  end

  subroutine arrays(ycen,ywid,work,mask,ncols,im,im2)

  c  --- copies ywid lines of the image and the mask into working arrays.
implicit none
integer cols
parameter (cols=2045)

integer ycen,ywid,ncols,i,j,ier,im,im2,mask(ncols,100)
real work(ncols,100), buf(cols)
do i=1,ywid
   call imgl2r (im, buf, ycen-ywid/2-1+i, ier)
do j=1,ncols
   work(j,i) = buf(j)
end do
call imgl2r (im2, buf, ycen-ywid/2-1+i, ier)
do j=1,ncols
   mask(j,i) = int(buf(j))
end do
d return
end

subroutine brick(ncols,nlines,hr,xc,ycen,ywid,
: work,mask,grid,firstwid,epdn)
c --- divides up a slice containing ywid rows into x-bricks,
c --- averages and writes out.
implicit none
integer ncols,nlines,xcen,xwid,firstwid,n,ycen
integer ywid, xc, i, xwidold, xstart, xend,xy(2,10000)
integer hr, xcenold, firstx, lastx,mask(ncols,100)
c --- EDM used brick width of 40 for 891 data
parameter (xwid=40,firstx =1)
real work(ncols,100), grid(ncols,nlines)
real buf(10000), val, err, truxc, truyc, epdn
lastx = ncols
xcen = xc
c --- CENTER BRICK (X):
c --- Tip the non-masked values into buf; n is no of values not masked.
call tip(xwid,ywid,xcen,ycen,mask,work,n,buf,xy,
: ncols,nlines,truxc,truyc)
c --- Work out 'average' of values in brick and write to estgal file.
call average(buf,xy,n,xwid,ywid,xcen,ycen,
: truxc,truyc,val,err,epdn)
c --- OTHER BRICKS ALONG X AXIS:
do i=1,2
95
xwidold = xwid
xcenold = xcen
10  call widcalc2(xc,xwidold,xcenold,i,xwid,xcen)

C     --- Need to check whether we have got to the end of the row.
     xstart = xcen - xwid/2
     xend = xstart + xwid -1
     if(xstart.ge.firstx.and.xend.le.lastx) then

C     --- Tip the non-masked values into buf; n is no of values not masked.
     call tip(xwid,ywid,xcen,ycen,mask,work,n,buf,xy,
        ncolsnlines,truxc,truyc)
     end if
C     --- Work out 'average' of values in brick and write to estgal file.
     call average(buf,xy,n,xwid,ywid,xcen,ycen,
        truxc,truyc,val,err,epdn)

C     --- Unless we have got to the end of the row, go back for another one.
     if(xstart.ne.firstx.and.xend.ne.lastx) then
        go to 10
     else
        xcen = xc
     end if
     else
        xcen = xc
     end if
     end do
     return
end

subroutine tip(xwid,ywid,xcen,ycen,mask,work,n,buf,xy,
   ncolsnlines,truxc,truyc)

C     --- tips the non-masked elements of the 'brick' into working array buf.
C     --- Also works out the true center of the non-masked pixels.

implicit none

integer ncolsnlines,xcoord,ycoord,xy(2,10000)
integer xwid,ywid,xcen,i,j,n,ycen,mask(ncols,100)
real work(ncols,100),buf(10000),truxc,truyc

n = 0
truxc = 0
truyc = 0
do i=1,xwid
   do j=1,ywid
      xcoord = xcen-xwid/2-1+i
      ycoord = ycen-ywid/2-1+j
      if(mask(xcoord,j) .ne.0) then
         n = n + mask(xcoord,j)
         buf(n) = work(xcoord,j)
      xy(i,n) = i
xy(2,n) = j
truxc = truxc + xcoord
truyc = truyc + ycoord
end if
end do
end do
if(n.ne.0) then
truxc = truxc/float(n)
truyc = truyc/float(n)
end if
return
end

subroutine average(buf,xy,n,xwid,ywid,xcen,ycen,: truxc,truyc,val,err,epdn)
c --- Given n values in buf, produces some sort of average and sensible
c --- estimate of error --- actual method depends on sample size.
c --- includes contributions from readout noise (ron), small-scale
c --- flat-fielding errors (sff), large-scale flat-fielding errors
c --- (lff), surface brightness fluctuations in galaxy itself (sbf),
c --- [and both small and large-scale sky variations (sskyvar and lskyvar)]
c --- EDM no sky variations for 891--simply double lff since it's
c --- the same effect
nc --- note if errors multiplicative

c --- also calculates actual sigma in this bin and compares with
nc --- theoretical estimate to see if we are on the right track.
c --- this involves getting rid of the actual variation of the galaxy
c --- across the patch of galaxy we are averaging
implicit none
real ron,sff,lff,sbf,sky

c --- for 4565 data:
c --- parameter(ron=1.6, sff=.00176, lff=.000852, sbf=0.36, sky=1760.)
c --- EDM for 891 data:
c --- parameter(ron=.218, sff=.000488, lff=.000728, sbf=0.36, sky=1877.)
integer i, j, n, xcen, ycen, ntrim, xwid, ywid
integer xstart, xend, ystart, yend,xy(2,10000),lista(3)
integer newxy(2,10000)
real buf(10000),val,err,mean,sig,trim,realtrim,epdn,a(3)
real trimmed(10000), order(10000), truxc, truyc, covar(3,3)
real white_noise, sbf_noise, sig2, sigerr, comperr,junk
real chisq, lqsq(10000),flag,logn,seeing
xstart = xcen - xwid/2
xend = xstart + xwid -1
ystart = ycen -ywid/2
yend = ystart + ywid -1

97
if(n.eq.0) then
    val = 0
    err = 0

c --- If only one value this is it and its poisson error
    (+ron+FF+sbf) is the error
else if(n.eq.1) then
    val = buf(1)

c --- EDM noise calculation now includes additional lff factor to compensate
    for sky variations
    white_noise = sqrt(val/float(n)/epdn + ron**2/float(n) +
            (sff*val)**2/float(n) + 2*(lff*val)**2 )
    if((val-sky).gt.0) then
        sbf_noise = sqrt(sbf*(val-sky)/n)
    else
        sbf_noise = 0
    end if
    err = sqrt(white_noise**2+sbf_noise**2)
write(1,10) truc, truc, val, err, n, xstart,
       xend, ystart, yend
10    format(4f10.2,5i6)

c --- If n==2-4 or we have a 3xn or nx3 brick (in which case there may
     well be real intensity variations across the brick because we are
     near the center of the galaxy), average them and work out
     what sigma is from their individual poisson errors.
else if(xwid.eq.3.or.(ywid.eq.3.and.xwid.lt.10).or.n.1e.4) then
    call stdev(buf,n,val,sig)
c --- EDM noise calculation now includes additional lff factor to compensate
    for sky variations
    white_noise = sqrt(val/float(n)/epdn + ron**2/float(n) +
            (sff*val)**2/float(n) + 2*(lff*val)**2 )
    if((val-sky).gt.0) then
        if(n.1e.700) then
            logn = max(alog10(float(n)),0.6)
            seeing = 6.72418 - 6.306444*logn + 2.416207*logn**2
            - 0.3174202*logn**3
        else
            seeing = 1
        end if
        sbf_noise = sqrt(sbf*(val-sky)/n)/seeing
    else
        sbf_noise = 0
    end if
    err = sqrt(white_noise**2+sbf_noise**2)
write(1,10) truc, truc, val, err, n, xstart,
       xend, ystart, yend

c --- If > 5 and xwid or ywid are >5, work out a trim sigma
    (10% trim, small sample trim one from top and bottom)
c --- Clip anything more than 3 of these sigma's away and then work
c --- out the mean and sigma with what is left. Theoretical
c --- error only.
else
call stddev(buf,n,val,sig2)

trim = 0.10
do i=1,n
  order(i) = buf(i)
end do
call sort(n,order)
realtrim = float(n)*trim
ntrim = jhint(realtrim) ! nearest integer
if(ntrim.eq.0) ntrim=1
do j=1,n-2*ntrim
  trimmed(j) = order(j+ntrim)
end do
call stddev(trimmed,n-2*ntrim,mean, sig)
sig = sig/0.661444

i = 0
do j=1,n
  if(abs(buf(j)-mean).le.2.5*sig) then
    i = i+1
    trimmed(i) = buf(j)
    newxy(1,i) = xy(1,j)
    newxy(2,i) = xy(2,j)
  end if
end do
call stddev(trimmed,i,val,sigerr)

c --- EDM noise calculation now includes additional lff factor to compensate

c --- for sky variations
    white_noise = sqrt(val/float(i)/epdn + ron**2/float(i) +
                   (sff*val)**2/float(n) + 2*(lff*val)**2 )
if((val-sky).gt.0) then
  if(i.le.700) then
    logn = max(aalog10(float(i)),0.6)
    seeing = 6.72418 - 6.306444*logn + 2.416207*logn**2
           - 0.3174202*logn**3
  else
    seeing = 1
  end if
  sbf_noise = sqrt(sbf*(val-sky)/i)/seeing
else
  sbf_noise = 0
end if
err = sqrt(white_noise**2+sbf_noise**2)

c --- need to get rid of large-scale ff error to compare with actual sigma
write(1,10) truxc, truyc, val, err, i, xstart,
:                     xend, ystart, yend
end if
return
end

subroutine stdev(x,n,mean,sig)
real x(10000), sig ,mean
real*8 sumx,sumxsq,variance
integer n
sumx = 0.
sumxsq = 0.
do i=1,n
   sumx = sumx + x(i)
   sumxsq = sumxsq + dble(x(i))**2
end do
mean = sngl(sumx)/float(n)
variance = (sumxsq - ((sumx)**2)/dble(n))/dble(n-1)
if(variance.ge.0) then
   sig = sqrt(sngl(variance))
else
   sig = 0
end if
return
end

subroutine widcalc(c,widold,cenold,h,k,wid,cen)
c --- Works out whether we have got far enough out from the center that
c --- it is safe to make the binsize bigger without risking significant
c --- changes in galaxy brightness in the bin.

implicit none

real change
parameter (change = 0.01)
integer c,widold,cenold,h,k,wid,cen,start,end
real*8 check
wid = widold
10 cen = cenold + (-1)**k*(widold+wid)/2
start = cen-wid/2
end = cen + wid/2
check = dexp(-abs(dfloat(start-c))/h) -
    dexp(-abs(dfloat(end-c))/h)
check = dabs(check)
if(sngl(check).le.change.and.wid.lt.99) then
    wid = wid + 2
    go to 10
end if
return
end
subroutine widcalc2(c, widold, cenold, k, wid, cen)
c --- dummy version of widcalc for x axis where xwid doesn't change
implicit none
integer c, widold, cenold, k, wid, cen, start, end
wid  = widold
  cen = cenold + (-1)**k*(widold+wid)/2
start = cen-wid/2
end = cen + wid/2
return
end
SUBROUTINE sort(n,arr)
INTEGER n,M,NSTACK
REAL arr(n)
PARAMETER (M=7,NSTACK=50)
INTEGER i,ir,j,jstack,k,l,istack(NSTACK)
REAL a,temp
jstack=0
1=1
ir=n
1 if(ir-1.lt.M) then
do 12 j=1+1,ir
   a=arr(j)
doi i=j-1,1,-1
   if(arr(i).le.a)goto 2
   arr(i+1)=arr(i)
11 continue
   i=0
2 arr(i+1)=a
12 continue
if(jstack.eq.0) return
ir=istack(jstack)
l=istack(jstack-1)
jstack=jstack-2
else
k=(l+ir)/2
temp=arr(k)
arr(k)=arr(l+1)
arr(l+1)=temp
if(arr(l+1).gt.arr(ir)) then
temp=arr(l+1)
arr(l+1)=arr(ir)
arr(ir)=temp
endif
if(arr(l).gt.arr(ir)) then
    temp=arr(l)
    arr(l)=arr(ir)
    arr(ir)=temp
endif
if(arr(l+1).gt.arr(l)) then
    temp=arr(l+1)
    arr(l+1)=arr(l)
    arr(l)=temp
endif
i=l+1
j=ir
a=arr(l)
3 continue
    i=i+1
    if(arr(i).lt.a) goto 3
4 continue
    j=j-1
    if(arr(j).gt.a) goto 4
    if(j.lt.i) goto 5
    temp=arr(i)
    arr(i)=arr(j)
    arr(j)=temp
    goto 3
5 arr(l)=arr(j)
arr(j)=a
jstack=jstack+2
if(jstack.gt.NSTACK) pause 'NSTACK too small in sort'
if(ir-i+1.ge.j-1) then
    istack(jstack)=ir
    istack(jstack-1)=i
    ir=j-1
else
    istack(jstack)=j-1
    istack(jstack-1)=l
    l=i
endif
endif
goto 1
END
C.6  magnitude

program magnitude
* magnitude calibration applied to perpendicular profiles

    implicit none
    integer i
    real truxc, truy, val, err, valss, highss, lowss
    real mag, highmag, lowmag
    character*80 infile, outfile

    write (*, '('' enter input file name: '',$)'))
    read (*, '(a80)') infile
    write (*, '('' enter output file name: '',$)'))
    read (*, '(a80)') outfile

    open(1,file=infile,status='old')
    open(2,file=outfile,status='new')

    do i=1,5000
        read(1,*end=20) truxc, truy, val, err
        valss = val - 1877
        highss = valss + err
        lowss = valss - err
        if(valss.le.0) valss = 0.01
        if(highss.le.0) highss = 0.01
        if(lowss.le.0) lowss = 0.01
        mag = 29.01 - 2.5*alog10(valss)
        highmag = 29.01 - 2.5*alog10(highss)
        lowmag = 29.01 - 2.5*alog10(lowss)
        write(2,10) truxc, truy, val, err, mag, highmag, lowmag
    10    format(7f10.2)
    end do

20    close(1)
    close(2)

end
References


van der Kruit, P.C. and Searle, L. *Astronomy and Astrophysics*. **95**: 105-115 (1981a)

van der Kruit, P.C. and Searle, L. *Astronomy and Astrophysics*. **95**: 116-126 (1981b)