

UCF Physics: AST 5765/4762:
(Advanced) Astronomical Data Analysis Lecture Notes

Prof. Joseph Harrington
Planetary Sciences Group
Department of Physics
University of Central Florida
Orlando, Florida 32816
jh@physics.ucf.edu

©2004 – 2018 Joseph Harrington. All rights reserved.

2 December 2019

Acknowledgements

These lecture notes developed from a similar course I taught in the IDL language at Cornell University. I adapted that course, in turn, from a course taught there by Prof. Steven Eikenberry (now at the University of Florida in Gainesville, Florida). Steve and I independently developed generally similar outlines for a course on this topic, he for the class he taught and I as part of a project in a course on Effective College Teaching at Cornell. When I took over the class after he left Cornell, I generally followed his syllabus but integrated many ideas from mine. Over the years, the level increased from sophomore to senior, content changed, the language switched to Python, and I added the graduate component. Still, the general outline remains similar to his. One of the projects, some of the homework datasets, and a few of the diagrams come from notes Steve was kind enough to share with me. As usual, I am grateful to him for sharing his materials. Any errors are mine. I continue to follow the tradition of awarding a homework problem's worth of credit to any student pointing out a substantive error in course materials.

Contents

Acknowledgements	i
1 Introduction	1
1.1 Check In : — :, 5 min	2
1.2 Initial Programming Assessment: : — :, 15 min	2
1.3 Course Introduction: : — :, 25 min	2
1.4 Research Integrity: : — :, 10 min	3
1.5 Lab Logs: : — :, 10 min	3
1.6 Student Introductions: : — :, 10 min	4
1.7 Computing Environment: : — :, 25 min	4
1.8 Emacs: : — :, 5 min	4
1.9 Linux Intro: : — :, 5 min	5
1.10 Check In : — :, 5 min	5
1.11 First Login: : — :, 15 min	5
1.12 System Management, Level 0 - Backup: : — :, 5 min	6
1.13 VirtualBox Management : : — :, 5 min	6
1.14 First Linux Play: : — :, 40 min	7
1.15 Check In : — :, 5 min	7
1.16 Computers and Data Analysis: : — :, 5 min	7
1.17 Computers and Data Analysis: : — :, 5 min	9
2 Learning Python	11
2.1 Python Intro: : — :, 15 min	11
2.2 Basic Python Demos: : — :, 30 min	12
2.3 Go Over HW1 — Peer Evaluation: : — :, 60 min	13
3 Python Data and Programming	15
3.1 Check In: : — :, 5 min	15
3.2 Python Basic Data Types: : — :, 20 min	15
3.3 Python Basic Container Data Types: : — :, 10 min	16
3.4 NumPy for Large Numerical Lists: : — :, 5 min	16
3.5 NumPy Basic Data Types: : — :, 10 min	16
3.6 Using Python Function Calls and Objects: : — :, 10 min	16
3.7 Argument Passing in Python: : — :, 10 min	17

4	Python Arrays and Image Display	19
4.1	Python Arrays: : — :, 25 min	19
4.2	Check In: : — :, 5 min	19
4.3	Astronomical Image Display: : — :, 15 min	20
5	Python Functions, Coding, FITS	21
5.1	HW2 Peer Evaluation: : — :, 60 min	21
5.2	Functions in General: : — :, 10 min	21
5.3	Using Git: : — :, 10 min	22
5.4	Python Functions: : — :, 10 min	22
5.5	Check In: : — :, 5 min	23
5.6	Arguments in Python: : — :, 15 min	23
5.7	Programming Style: : — :, 15 min	24
5.8	Astronomical Data: : — :, 25 min	25
6	Measurement, Probability, and Statistics	27
6.1	Illegitimate, Systematic, and Random Errors: : — :, 10 min	27
6.2	What's in a Measurement?: : — :, 10 min	28
6.3	Terms: : — :, 10 min	28
6.4	Testing Theory: : — :, 10 min	29
6.5	Probability Basics: : — :, 20 min	29
6.6	Estimates of Mean, etc.: : — :, 5 min	30
7	Probability Distributions and Error Analysis	31
7.1	Check In: : — :, 5 min	31
7.2	Gaussian Distribution: : — :, 10 min	31
7.3	Poisson Distribution: : — :, 15 min	32
7.4	Confidence Intervals: : — :, 5 min	33
7.5	Uncertainty of a Measurement: : — :, 5 min	33
7.6	Goal of Error Analysis: : — :, 5 min	33
7.7	Error Propagation Equation: : — :, 15 min	33
7.8	HW3 Peer Evaluation: : — :, 60 min	35
7.9	Check In: : — :, 5 min	35
7.10	Error in Addition: : — :, 10 min	35
7.11	Error in Multiplication: : — :, 10 min	36
7.12	Error in Mean and Wrapup: : — :, 10 min	36
7.13	HW4 Peer Evaluation: : — :, 50 min	37
7.14	Computational Error Analysis (Monte Carlo): : — :, 5 min	37
8	Signal-to-Noise Ratio Calculations	39
8.1	Calculating Astronomical Signal-to-Noise Ratios: : — :, 5 min	39
8.2	Simple Scaling Rules: : — :, 5 min	39
8.3	Calculating Signal: : — :, 5 min	40
8.4	Calculating Noise: : — :, 10 min	40
8.5	Signature of Unknown Errors: : — :, 5 min	40

8.6	Practice: Exoplanet S/N: : — :, 10 min	41
8.7	Quiz 1: : — :, 30 min	41
9	Fitting	43
9.1	Check In: : — :, 10 min	43
9.2	Fitting Basics: : — :, 10 min	43
9.3	Least Squares Linear Fitting: : — :, 25 min	44
9.4	Function Minimization and Goodness of Fit: : — :, 10+5 min	45
9.5	HW5 Peer Evaluation: : — :, 60 min	45
10	Astronomical Measurements, CCD Arrays, Array Correction	47
10.1	Check In: : — :, 5 min	47
10.2	Types of Astronomical Measurements: : — :, 5 min	47
10.3	What's After the Telescope?: : — :, 15 min	47
10.4	Array Basics: : — :, 10 min	48
10.5	From Photon to Data: : — :, 20 min	48
10.6	CCD Characteristics and Systematics: : — :, 15 min	49
10.7	Check In: : — :, 5 min	50
10.8	Summary of CCD Systematics: : — :, 5 min	50
10.9	Basic CCD Error Correction: : — :, 5 min	51
10.10	Making the Calibration Frames: : — :, 10 min	51
11	Infrared Arrays, Sky, and Corrections	53
11.1	Infrared Arrays: : — :, 20 min	53
11.2	Characteristics of Infrared Arrays: : — :, 5 min	54
11.3	Near-IR Sky: : — :, 10 min	54
11.4	Mid-IR Sky: : — :, 10 min	55
11.5	Array Effects: : — :, 10 min	55
11.6	HW6 Peer Evaluation: : — :, 60 min	56
12	Combining Frames	57
12.1	Check In: : — :, 5 min	57
12.2	Flats: : — :, 5 min	57
12.3	Median Combination: : — :, 40 min	57
12.4	Cleaning and Producing Sky Flats: : — :, 10 min	58
12.5	Coding an Analysis: : — :, 10 min	59
12.6	Software Implementation: : — :, 10 min	59
12.7	Quiz 2: : — :, 20 min	60
12.8	FITS Implementation: : — :, 20 min	60
13	PSF and Photometry	63
13.1	Check In: : — :, 10 min	63
13.2	\$N\$-Dimensional Gaussian: : — :, 10 min	63
13.3	Convolution: : — :, 10 min	64
13.4	HW7 Peer Evaluation: : — :, 60 min	64

13.5	Point-Spread Function (PSF): : — :, 20 min	64
13.6	Airy diffraction Pattern: : — :, 10 min	65
13.7	Centering: : — :, 5 min	66
14	Aperture Photometry	67
14.1	Data Formation Model (“A Star is Born...In Python”): : — :, 5 min	67
14.2	Choosing the Star Aperture and Sky Annulus: : — :, 5 min	67
14.3	Aperture Photometry Formula: : — :, 5 min	68
14.4	Apertures in Python: : — :, 10 min	69
14.5	Or Else!: : — :, 5 min	69
14.6	Advanced 1: Aperture Correction: : — :, 5 min	69
14.7	Advanced 2: Interpolated Aperture Photometry: : — :, 5 min	70
15	Bad Pixels	71
15.1	Check In: : — :, 10 min	71
15.2	HW8 Peer Evaluation: : — :, 60 min	71
15.3	Bad Pixels and Cosmic Rays: : — :, 10 min	71
15.4	Making Masks: : — :, 15 min	72
15.5	Open Lab, Work on Project: : — :, 60 min	73
15.6	Open Lab, Work on Project With Instructor Help: : — :, 45 min	73
16	Extended-Source Photometry and Atmospheric Calibration	75
16.1	Check In: : — :, 10 min	75
16.2	Extended-Source Photometry: : — :, 10 min	75
16.3	Photometric Calibration: : — :, 5 min	76
16.4	The Absolute Calibration Bootstrap: : — :, 10 min	76
16.5	The Magnitude System: : — :, 10 min	77
16.6	Standard Star Calibration: : — :, 5 min	77
16.7	Atmospheric Correction: : — :, 10 min	78
16.8	Common Calibration Mistakes: : — :, 5 min	79
17	The Project	81
17.1	The Project: : — :, 15 min	81
17.2	Project Grading: : — :, 10 min	82
18	Interpolation and Correlation	85
18.1	Check In: : — :, 10 min	85
18.2	Interpolation: : — :, 15 min	85
18.3	Cross Correlation: : — :, 45 min	86
18.4	Coding Notes: : — :, 20 min	87
19	Introduction to Spectroscopy	89
19.1	Check In: : — :, 5 min	89
19.2	What is Spectroscopy?: : — :, 10 min	89
19.3	Spectroscopy Basics: : — :, 5 min	89
19.4	Overview of Spectroscopy: : — :, 25 min	90

19.5 Quiz 3: : — ;, 20+10 min	90
19.6 Go Over HW9: : — ;, 45 min	90
19.7 Spectral Observing Modes: : — ;, 5 min	90
19.8 Spectroscopy Examples: : — ;, 5 min	91
20 Spectroscopic Gratings and Extraction	93
20.1 Check In: : — ;, 5 min	93
20.2 Spectrograph Optics: : — ;, 25 min	93
20.3 Slit Width and Resolving Power: : — ;, 5 min	93
20.4 Long-Slit Spectroscopy: : — ;, 35 min	94
20.5 Gratings: : — ;, 10 min	95
21 Spectroscopic Systematic Corrections	97
21.1 Check In: : — ;, 5 min	97
21.2 Things to Correct: : — ;, 5 min	97
21.3 How to Correct Them: : — ;, 5 min	97
21.4 Frames to Acquire: : — ;, 5 min	98
22 Spectroscopic Wavelength Calibration	99
22.1 Wavelength Calibration Basics: : — ;, 25 min	99
22.2 Example Using OH ⁻ Lines: : — ;, 30 min	100
22.3 HW10 Peer Evaluation: : — ;, 45 min	100
22.4 Calibration Sources: : — ;, 10 min	100
22.5 Calibration Observations: : — ;, 5 min	100
23 Spectroscopy Applications	101
23.1 GTA Evaluation: 10:05 — 10:15, 10 min	101
23.2 Check In: : — ;, 10 min	101
23.3 Gravitational Redshift: : — ;, 5 min	101
23.4 Doppler Shift: : — ;, 5 min	101
23.5 Spectral Line Profiles: : — ;, 5 min	102
23.6 Electrical and Zeeman Broadening: : — ;, 5 min	102
23.7 Temperature Broadening: : — ;, 5 min	102
23.8 Lifetime/Collisional Broadening: : — ;, 10 min	103
23.9 Dark Matter: : — ;, 15 min	103
24 The Project Paper	105
24.1 Paper Content : — ;, 10 min	105
24.2 Paper Format : — ;, 10 min	105
25 Probability and Error Analysis Review	107
25.1 Selection of Review Topics: : — ;, 10 min	107
25.2 Review: : — ;, 90 min	107
26 Detector Physics Review	109

27 General Model Fitting	111
27.1 Check In: : — :, 10 min	111
27.2 Using Models to Interpret Data: : — :, 5 min	111
27.3 Least Squares: : — :, 10 min	112
27.4 Project One-on-One Help: : — :, 55 min	112
28 Optimal Extraction	113
28.1 Optimal Photometry: : — :, 30 min	113
29 Presenting Data: Plotting Well	115
29.1 Check In: : — :, 5 min	115
29.2 Video by Hans Rosling: : — :, 25 min	115
29.3 Discussion of Video by Hans Rosling: : — :, 20 min	115
29.4 Quiz 3: : — :, 20+10 min	116
29.5 Rules for Plots: : — :, 25 min	116
29.6 Your Work: : — :, 15+5 min	117
30 Presenting Data: Writing a Paper	119
30.1 Check In: : — :, 5 min	119
30.2 Learning to Write Research Papers: : — :, 5 min	119
30.3 Discussion of Deming <i>et al.</i> (2005): : — :, 20 min	120
30.4 Parts of a Research Paper: : — :, 15 min	120
30.5 The Abstract: : — :, 5 min	121
30.6 The Introduction: : — :, 10 min	121
30.7 The Main Text: : — :, 10 min	122
30.8 The Conclusions: : — :, 10 min	122
30.9 Paper Writing Style: : — :, 10 min	122
30.10 Discuss Papers Class Read: : — :, 20 min	123
31 Bayesian Statistics and Markov Chains	125
31.1 Two Approaches to Statistics: : — :, 5 min	125
31.2 Conditional Probability: : — :, 10 min	125
31.3 Bayesian Statistics: : — :, 10 min	126
31.4 Mapping Phase Space: : — :, 5 min	127
31.5 Stevenson MCMC Demo: : — :, 25 min	127
32 Postmortem	129
32.1 Check In: : — :, 5 min	129
32.2 Quiz 6: : — :, 25 min	129
32.3 Class Postmortem: : — :, 20 min	129
33 Random Sections	133
33.1 Maskinterp: : — :, 10 min	133
33.2 Maskinterp Demo: : — :, 5 min	133
33.3 General Linear Fitting: : — :, 10+5 min	133

34 Periodic Signals and Power Spectra	135
34.1 Check In: : — :, 10 min	135
34.2 What is a Power Spectrum?: : — :, 5 min	135
34.3 Goals of Calculating the Power Spectrum: : — :, 5 min	136
34.4 Orthonormal Basis Functions: : — :, 10 min	136
34.5 Transforms: : — :, 5 min	137
34.6 Sound Example: : — :, 10 min	137
34.7 Properties: : — :, 10 min	138
34.8 Real Data, Sampling, and Aliasing: : — :, 10 min	138
34.9 Fourier Transform: : — :, 10 min	139
34.10 Discrete Fourier Transform: : — :, 10 min	140
34.11 DFT Speed: : — :, 5 min	140
34.12 Amplitude, Phase, Power Spectrum in Python: : — :, 50 min	141
34.13 FFT Algorithm: : — :, 10 min	141
34.14 FFT Speed: : — :, 5 min	142
35 Fourier Transform Applications	143
35.1 Check In: : — :, 10 min	143
35.2 HW11 Peer Evaluation: : — :, 45 min	143
35.3 Waveforms and Harmonics: : — :, 10 min	143
35.4 Periodograms: : — :, 15 min	143
35.5 Estimating a Period With the FFT: : — :, 10 min	145
35.6 Wavelets: : — :, 45 min	145

Chapter 1

Introduction

Pre-Class:

- Email students long before they reach class that they need to know how to program and have a decent computer
- Email students before the first class to download the Virtualbox, Ubuntu, and Anaconda installers
- Make a new VirtualBox for this year, update handout
- Prepare WebCourses files, syllabus, and initial HW
- Prepare .bash`aliases, including Python variables
- Prepare TA
- Test students' dotfiles, confirm Python path gets our stuff, test Python
- Copy handouts, demos, visuals to WebCourses

Handouts:

- Syllabus
- Learning Linux
- Learning Python
- emacs-cheat-sheet
- FWUnix reference sheet
- Research_Integrity_Guidelines_UCF.pdf
- comp`env`setup
- HW1
- HW2
- lecture notes 1
- lecture notes 2

Class time is critical throughout the course, but particularly in the early weeks. Students must get set up right away with VirtualBox on their laptops so that they can access the course computing environment. So, do the most basic of introductions, get them started installing, and have the TA send them to the system manager. Expect to spend a full class period getting VBox, Linux, and Python set up on their machines. They can do the first assignment with just Linux. In the first three classes, introduce in the first, spend the second installing VNC, and make the first homework due on the third.

1.1 Check In : — :, 5 min

- Questions before we start?

1.2 Initial Programming Assessment: : — :, 15 min

1.3 Course Introduction: : — :, 25 min

- Video-recording lectures this year
- Introduce me, staff, and class topic
- Ask if everyone can bring a laptop
- Disability accommodations
- Handouts, lecture notes, homework, Python stuff, data in WebCourses
- Split grad/ugrad class
- Review schedule and course approach/organization
- Set office hours with whenisgood
- Review syllabus and homework instructions
- Discuss peer evaluation
- Books
 - *An Introduction to Data Analysis for the Physical Sciences* by Bevington and Robinson
 - *Handbook of CCD Astronomy* by Howell
 - *Numerical Recipes* by Press *et al.*
 - *Python Scripting for Computational Science* by Langangen
 - NumPy and SciPy manuals
 - *Guide to Numpy* by Oliphant
 - *Using Python for Interactive Data Analysis* by Greenfield and Jędrzejewski
 - Learning Python: You pick based on your background
 - * byteofpython.com for those who can program
 - * diveintopython.com a little more depth
 - * *Learning to Program Using Python* by Gauld
 - * *Internet Programming in Python* by Watters, van Rossum, and Ahlstrom
 - * *Programming Python* by Lutz
 - * Python web sites, indexed by wikipedia.org article
 - * I want your reviews and suggestions

1.4 Research Integrity: : — :, 10 min

- Research integrity is a major national concern
- <http://www.research.ucf.edu/compliance.html>
- <http://www.policies.ucf.edu/documents/4-211ResearchMisconductPolicy.pdf>
- Need to be honest and as unbiased as possible
- Not put personal interests (including career advancement, compensation) before interests of humanity or the sponsor
- Not misappropriate funds, have conflicts of interest, falsify data, lie about results, etc.
- *Must provide documentation of all research actions, decisions, results*
- NSF, NASA, and other Federal agencies are now auditing much more often
- Very serious consequences against future career and employer even just for getting investigated (reputation forever in question by peers, negative media attention)
- If found guilty, fines, disbarment, jail time for you and your supervisor or institution

1.5 Lab Logs: : — :, 10 min

- Purpose of lab log: make a record of what you did for you and others
- Mostly useful to you and your supervisor
- Logs are the main item scrutinized by integrity audits
- Seriousness of lab logs in commercial research, Federal audits
- Other purposes
 - Online, searchable, can date-match to email
 - Used to save you time in the future
 - Work in your lab log, cut and paste
- Start entry at bottom with “***** IN: ” and output of date, like this:
- Leave blank line above and below
- Never (EVER) change above current date line. Be able to swear to this in court!
- If tracking time, end with “***** OUT: ” and output of date
- Here is a sample entry:

***** IN: Tue Aug 21 09:17:35 EDT 2018

Today, I started a work log for AST 5765 Advanced Astronomical Data Analysis. I took some notes:

Don't cheat on research! Jail is no fun.
Don't forget to put nameplate up at start of each class and start VirtualBox.

DO:

login to WebCourses and explore all class areas
read syllabus and all handouts from Files area
install class software and attempt homework 1
Do by Thu for extra credit and for Thu lecture

***** OUT: Tue Aug 21 09:17:35 EDT 2018

1.6 Student Introductions: : — :, 10 min

- Nameplates (hand out paper)
- 30 seconds to say name, year, major, research, something interesting and non-school

1.7 Computing Environment: : — :, 25 min

- Our computing environment is a little complicated to set up
- Install it on your machine so you can compute after course, learn on your own (and have class in normal classrooms on weekdays...)
- No bottlenecks when large tasks assigned
- Start now, follow handout, finish *tonight*, extra credit if done by next class
- Help others get set up
- Come to us for help

1.8 Emacs: : — :, 5 min

- Emacs intro: Find lab log
 - Start emacs in background
 - Find lab log
 - Type, delete, copy, cut, paste, save, quit

- Multiple emacs buffers, buffer list
- Emacs tutorial
- Emacs HOWTO
- emacs-cheat-sheet
- Start new Python in emacs

1.9 Linux Intro: : — :, 5 min

- Why Linux?
 - Main scientific number-crunching environment in cloud and research labs
 - Easy to do anything computer can do
 - GUIs for most things, shell if you need it
 - Free
 - In wide use in astronomy and physical sciences
 - For us, it's a means to an end. Pipelines use it more extensively.
 - Often need to do unusual things (“If we knew what it was that we were doing, it wouldn't be research.”)
 - Linux and most programs running on it come with source code and supportive user communities

1.10 Check In : — :, 5 min

- Questions before we start?

1.11 First Login: : — :, 15 min

- We're going to login and get a command prompt now
- Warning: Today will be mainly system setup. Normal classes won't be like this!
- If you are stuck or something isn't as expected, ASK FOR HELP! This is important if we are to do this in a reasonable amount of time. Do NOT be shy! We only bite on Wednesdays.
- Get a command line in a terminal from the start menu
- Let the lecturer know you got this far

1.12 System Management, Level 0 - Backup: : — :, 5 min

- Everyone learns the lesson of backup
- **It is better to learn it from somebody else's pain!**
- You can easily follow the install instructions to make a second VBox with our software.
- You can't cause your deleted home directory to reappear!
- Tar up your home directory and copy that file to your laptop
- Copy it elsewhere, too - Google Drive, home computer, flash drive
- Copy your `ast5765` directory often, and do the same (quick, small file).

1.13 VirtualBox Management : : — :, 5 min

- VirtualBox really is like having a separate computer in your computer
- It can be very unforgiving!
- It can also be slow, as it makes no use of graphics hardware. Everything is done in software.
- Configure as much memory as you can tolerate. If your computer has 4GB, configure at least 3GB for VirtualBox.
- Use `htop` in a terminal to monitor both the user (green) and system (red) CPU, memory, and swap space use.
- If the CPU is completely used, the memory is not full, and swap space isn't used much, it's going as fast as it can on your machine.
- If the memory is full and it's using swap space, configure more memory.
- If the swap space is full, that's trouble. Configure more swap space. **DO NOT** shrink the filesystem partition! Increase the size of the whole disk, and add that to the swap space. Ask us for help if needed.
- If things are running slowly or you have other trouble
 - Wait out any problem. At worst, reboot.
 - Backup your `ast5765` directory
 - Take a snapshot of your VBox
 - Come to Katie or jh for help. Michael and Zaccy also.
 - Do not delete your swap space, resize partition, or do other recipes found online!
 - Did I mention backup? Saves *hours*, sometimes days.

1.14 First Linux Play: : — :, 40 min

- Demo: `01shell`
- What is a shell? A command-line program for running other programs
- Handout: `ls`, `cd`, `cat`, `more/less`, `mkdir`, `rmdir`, `cp -a`, `date`, `rm`
- Variables: shell and environment
- `$PATH` variable, `/usr/local/bin`
- aliases and `alias rm='rm -i'`
- `.bashrc`
- Documentation: `help.ubuntu.com`, `man`, `info`, `/usr/share/doc`, `web`
- Explore directory structure, make “handin” directory
- `Stdin`, `stdout`, `stderr`, redirection, pipes

1.15 Check In : — :, 5 min

- Questions before we start?
- Who read the handout on learning Linux?
- Does everyone have a terminal started and the demo ready?

While the next section seems like an intro that should have come earlier, the need to get the students functional in the computing environment puts it at the end of this chapter.

1.16 Computers and Data Analysis: : — :, 5 min

- Example of useful shell command line that you can’t do in Windows. I want to update the syllabus so that the topics reflect what I actually did last year. Each directory contains the LaTeX outline for the lecture, with a title giving the topic. In a file browser, I could click down to each of 29 directories, double-click on the lecture PDF, and read the title. In the shell, I did this:

```
jh@glup ~/ast/teaching/ast5765/past/lectures $ ls
4 ast5765-L01-2016-08-23-T/      4 ast5765-L16-2016-10-13-R/
4 ast5765-L02-2016-08-25-R/      4 ast5765-L17-2016-10-18-T/
4 ast5765-L03-2016-08-30-T/      4 ast5765-L18-2016-10-20-R/
4 ast5765-L04-2016-09-01-R/      4 ast5765-L19-2016-10-25-T/
4 ast5765-L05-2016-09-06-T/      4 ast5765-L20-2016-10-27-R/
```

```

4 ast5765-L06-2016-09-08-R/      4 ast5765-L21-2016-11-01-T/
4 ast5765-L07-2016-09-13-T/    4 ast5765-L22-2016-11-03-R/
4 ast5765-L08-2016-09-15-R/    4 ast5765-L23-2016-11-08-T/
4 ast5765-L09-2016-09-20-T/    4 ast5765-L24-2016-11-10-R/
4 ast5765-L10-2016-09-22-R/    4 ast5765-L25-2016-11-15-T/
4 ast5765-L11-2016-09-27-T/    4 ast5765-L26-2016-11-17-R/
4 ast5765-L12-2016-09-29-R/    4 ast5765-L27-2016-11-22-T/
4 ast5765-L13-2016-10-04-T/    4 ast5765-L28-2016-11-29-T/
0 ast5765-L14-2016-10-06-Rcxl  4 ast5765-L29-2016-12-01-R/
4 ast5765-L15-2016-10-11-T/    4 lectall/
jh@glup ~/ast/teaching/ast5765/past/lectures $ grep Lecture a*/a*.tex
ast5765-L01-2016-08-23-T/ast5765-L01-2016-08-23-T.tex:{\large\bfseries
ast5765-L01-2016-08-23-T/ast5765-L01-2016-08-23-T.tex:\comment{*** Lect
ast5765-L02-2016-08-25-R/ast5765-L02-2016-08-25-R.tex:{\large\bfseries
ast5765-L03-2016-08-30-T/ast5765-L03-2016-08-30-T.tex:{\large\bfseries
ast5765-L04-2016-09-01-R/ast5765-L04-2016-09-01-R.tex:{\large\bfseries
ast5765-L05-2016-09-06-T/ast5765-L05-2016-09-06-T.tex:{\large\bfseries
ast5765-L06-2016-09-08-R/ast5765-L06-2016-09-08-R.tex:{\large\bfseries
ast5765-L07-2016-09-13-T/ast5765-L07-2016-09-13-T.tex:{\large\bfseries
ast5765-L08-2016-09-15-R/ast5765-L08-2016-09-15-R.tex:{\large\bfseries
ast5765-L09-2016-09-20-T/ast5765-L09-2016-09-20-T.tex:{\large\bfseries
ast5765-L10-2016-09-22-R/ast5765-L10-2016-09-22-R.tex:{\large\bfseries
ast5765-L11-2016-09-27-T/ast5765-L11-2016-09-27-T.tex:{\large\bfseries
ast5765-L12-2016-09-29-R/ast5765-L12-2016-09-29-R.tex:{\large\bfseries
ast5765-L13-2016-10-04-T/ast5765-L13-2016-10-04-T.tex:{\large\bfseries
ast5765-L15-2016-10-11-T/ast5765-L15-2016-10-11-T.tex:{\large\bfseries
ast5765-L16-2016-10-13-R/ast5765-L16-2016-10-13-R.tex:{\large\bfseries
ast5765-L17-2016-10-18-T/ast5765-L17-2016-10-18-T.tex:{\large\bfseries
ast5765-L18-2016-10-20-R/ast5765-L18-2016-10-20-R.tex:{\large\bfseries
ast5765-L19-2016-10-25-T/ast5765-L19-2016-10-25-T.tex:{\large\bfseries
ast5765-L20-2016-10-27-R/ast5765-L20-2016-10-27-R.tex:{\large\bfseries
ast5765-L21-2016-11-01-T/ast5765-L21-2016-11-01-T.tex:{\large\bfseries
ast5765-L22-2016-11-03-R/ast5765-L22-2016-11-03-R.tex:{\large\bfseries
ast5765-L23-2016-11-08-T/ast5765-L23-2016-11-08-T.tex:{\large\bfseries
ast5765-L24-2016-11-10-R/ast5765-L24-2016-11-10-R.tex:{\large\bfseries
ast5765-L25-2016-11-15-T/ast5765-L25-2016-11-15-T.tex:{\large\bfseries
ast5765-L26-2016-11-17-R/ast5765-L26-2016-11-17-R.tex:{\large\bfseries
ast5765-L27-2016-11-22-T/ast5765-L27-2016-11-22-T.tex:{\large\bfseries
ast5765-L28-2016-11-29-T/ast5765-L28-2016-11-29-T.tex:{\large\bfseries
ast5765-L29-2016-12-01-R/ast5765-L29-2016-12-01-R.tex:{\large\bfseries

```

So, in one command I have what I want. It's too wide, so I trimmed it down with a sed command.

```

jh@glup ~/ast/teaching/ast5765/past/lectures $ grep Lecture a*/a*.tex |

```

```

1, Tue 2016 Aug 23: Introduction
2}
2, Thu 2016 Aug 25: Learning Linux and Python
3, Tue 2016 Aug 30: Python Data and Programming
4, Thu 2016 Sep 1: Python Data and Programming
5, Tue 2016 Sep 6: Python Arrays and Image Display
6, Thu 2016 Sep 8: Functions, Coding Style,
7, Thu 2016 Sep 8: Measurement, Probability,
8, Thu 2016 Sep 14: Probability Distributions
9, Tue 2016 Sep 20: Fitting
10, Thu 2016 Sep 22: Fitting and Astronomical
11, Tue 2016 Sep 27: CCDs
12, Thu 2016 Sep 29: Infrared Arrays, Sky,
13, Tue 2016 Oct 4: Infrared
15, Tue 2016 Oct 1: Coding, FITS,
16, Thu 2016 Oct 12: FITS, Point-Spread Function
17 Tue 2016 Oct 18: Presenting Data
18, Thu 2016 Oct 20: Presenting Research:
19, Tue 2016 Oct 25: Project, PSF, Centering
20, Thu 2016 Oct 27: Aperture Photometry
21, Tue 2016 Nov 1: Bad Pixels
22, Thu 2016 Nov 3: Coding, Spectroscopy Intro
22, Thu 2016 Nov 3: Spectroscopy
24, Thu 2016 Nov 10: Spectroscopy: Gratings
25, Tue 2016 Nov 15: Interpolation and Correlation
26, Thu 2016 Nov 17: Spectroscopic Wavelength Calibration
27, Tue 2016 Nov 22: Spectroscopy Applications
28, Tue 2016 Nov 29: Bayesian Statistics and
29, Thu 2016 Dec 1: Optimal Extraction

```

The 2} on the second line came from a second line containing the word “Lecture” in the first lecture file.

This is the information I wanted, in a compact format, obtained in a fraction of the time it would have taken in a GUI file browser. Once you have good facility with the shell, this kind of thing happens a lot, many times an hour.

1.17 Computers and Data Analysis: : — :, 5 min

- Computer: powerful, integrated collection of data storage, memory, input/output, and calculation devices
- Today most interaction is via GUIs
- GUIs present interface to computer that makes sense for application

- GUIs very hard to write correctly
- Generally good for “canned” operations
- If user is doing new things, GUI is almost impossible
- Data analysis is in this category

Graphical User Interface (GUI)
Interpreted (text, no details, interactive, shell/IDL/Python)
Compiled (text, all details, C/C++/Fortran)
Assembly language (mnemonic)
Binary machine code
Central Processing Unit (CPU)

- GUI and command-line progs all written in compiled & interpreted languages
- For data analysis, we work in interactive (i.e., command prompt) languages that access compiled routines
- Fast, totally flexible
- Much easier to learn and use than C
- Lots of standard analysis methods already written
- But, must roll up sleeves and learn to program
- We use Python to manipulate data
- `emacs` to edit, run, and debug Python programs and logs
- Linux as OS

Chapter 2

Learning Python

2.1 Python Intro: : — :, 15 min

- Why Python?
- Python is an interactive (i.e., command prompt) language that accesses fast routines written in compiled languages like C
- Fast, totally flexible
- Hides many details: much easier to learn and use than C
- In astronomy, data are often arranged in arrays or lists

– Python is a high-level array language:

C:

```
#include <fitsio.h>      /* FITS file I/O package */
#include <memalloc.h>    /* memory allocation package */
double **a;
double **b;
double **c;
char **ahead;
char **bhead;
int i;
int j;
int ni;
int nj;

a = readfits("a.fits", ahead);
b = readfits("b.fits", bhead);
ni = sxpar(ahead, "NAXIS1"); /* assume same size */
nj = sxpar(ahead, "NAXIS2"); /* better would be to check! */
c = double2Dalloc(nj, ni);
```

```

for (j = 0; j < nj; j++){
  for (i = 0; i < ni; i++){
    c[j][i] = a[j][i] * b[j][i];
  }
}

```

Python:

```

import astropy.io.fits
a = astropy.io.fits.getdata('a.fits')
b = astropy.io.fits.getdata('b.fits')
c = a * b

```

- No declarations, language takes care of memory management
- Science and astronomy routines
- In wide use, replacing IDL
- emacs puts you in Python mode
- Syntax highlighting, indenting
- Python help: language
- Python.org, python manual and tutorial
- *ThinkPython* — rank beginner
- *A Byte of Python* — intermediate
- *Dive into Python* — expert programmer
- `help()`
- Python help: SciPy
- scipy.org cookbook, functional list of routines
- *Guide to NumPy*
- `scipy-user@scipy.org` mailing list
- SciPy doc wiki: `docs.scipy.org`
- Second homework is Python practice, third is writing programs.

2.2 Basic Python Demos: : — :, 30 min

- Demo: `02pythonintro.py`
- Quick demo from Matplotlib tutorial

2.3 Go Over HW1 — Peer Evaluation: : — :, 60 min

- Staff makes HW available for peer evaluation
- Staff puts list of students on screen/board in username order
- Students find their username in the list. Evaluate the next person down (this week).
- Students download the eval HW set to their `~/peer/hw1` directory
- Staff goes around class confirming each student's peer evaluatee and directory setup
- Students copy downloaded directory to `runtest`
- `cd` into `runtest`, delete output files
- Staff puts rubric on screen, steps students through it
- Students go through HW, executing commands and referencing solutions
- Students make comments in rubric
- Students finish evaluating by noon the following day
- Staff uses filled-in rubric as advice in grading

Chapter 3

Python Data and Programming

3.1 Check In: : — :, 5 min

- Questions before we start?
- Who went over the demo from last time at home?
- Who is comfortable working in Linux at this point?
- Who read the handout on learning Python?
- Does everyone have a terminal started and the demo ready?
- How is HW2? Anyone done?
- Examples/templates in python directory

3.2 Python Basic Data Types: : — :, 20 min

- Demo: `03pythonbasics.py`
- What is a variable?
- `string`, `int`, `float`
- `None` and `type()`
- In-class exercises:
 - What is the largest number Python can hold?
 - What is the smallest positive number Python can hold? On your own:
 - What is the most-negative number Python can hold?
 - What is the largest negative number (closest to 0) Python can hold?
- Basic operations, precedence, promotion

- Precision of printed output (1. - 0.3)
- Integer division in Python 2 and other languages, Python 3
- (Leave overflow, special values to Numpy intro)
- Numpy supports many more types and precisions

3.3 Python Basic Container Data Types: : — :, 10 min

- Lists
- Tuples
- Dictionaries

3.4 NumPy for Large Numerical Lists: : — :, 5 min

- range (slow!)
- xrange (fast but sequential)
- arange (fast and random)

3.5 NumPy Basic Data Types: : — :, 10 min

- The possible types are `int8`, `uint8`, `int16`, `uint16`, `int32`, `uint32`, `int64`, `uint64`, `float32`, `float64`, `complex64`, `complex128`.
- Using `int`, `float`, `complex` gets the default precision for that type.
- Precision of printed output
- Overflow and exceptional values: NaN, Inf, -Inf

3.6 Using Python Function Calls and Objects: : — :, 10 min

- Function calls have return values (even if None)
- You can ignore the return value or use it
- Arguments are in parens, comma-separated
- Positional args are first, then any keyword arguments (random order)
- Object: collection of data items and functions (called methods) under 1 name

- How to get an object
- Object data
- Calling methods (often no arg list)
- Everything is by reference: assignment, function call args

3.7 Argument Passing in Python: : — :, 10 min

- Transferring data from outside to inside a routine is called argument passing
- “Pass by value” makes a copy of the data
- Can modify all you want, does not affect the caller
- But what if the data are a huge array taking up 3/4 of the memory? Failure!
- “Pass by reference” copies a *pointer* to the data
- Saves memory, but now changes made to data affect the caller afterward
- “Object passing” is pass by reference, but most operations make new objects (by copying) so it acts like pass by value
- Use in-place operations to avoid making copies (`/=`, `a.flat[:]=`)
- Programmers generally assume routines do not affect data in the caller
- Do not modify inputs unless you intend that behavior
- Document it if you do
- Otherwise, copy inputs before modifying!

Chapter 4

Python Arrays and Image Display

4.1 Python Arrays: : — :, 25 min

- Array allocation
- Array slicing
- Assigning into an array
- Boolean and mask arrays
- Index initialization
- `.shape`
- Multidimensional arrays
- Linear array access

4.2 Check In: : — :, 5 min

- Questions before we start?
- Do you have a terminal running python, and the lecture demo ready to go?
- Are people getting comfortable with Linux and Python?
- Examples/templates in python directory
- How long did HW2 take?
- Update occasionally:
`conda update --prefix --all`
`conda update --prefix ~/miniconda3/envs/astroconda --all`
`apt update ; apt upgrade` (NOTE: In a virtual environment, it's not as critical, but still a good idea.)

4.3 Astronomical Image Display: : — :, 15 min

- Demo: `04imdispfunc.py`
- A pixel is colored by three one-byte values, for Red, Green, and Blue
- Unsigned one-byte integers range from 0-255 (8 binary 1 bits)
- Astronomical images and spectra are usually 2D arrays, 2-4 bytes per pixel
- Some data have a time and/or wavelength dimension as well
- Viewing astronomical images requires scaling and color mapping
- Linear scaling formula:

$$A'_{ij} = 255 \times \frac{A_{ij} - \min(A)}{\max(A) - \min(A)} \quad (4.1)$$

- There are also log scalings, sqrt, histeq, and so on
- Colormaps (including gray scale)
- ds9 program

Chapter 5

Python Functions, Coding, FITS

5.1 HW2 Peer Evaluation: : — :, 60 min

NOTE: Skip if class can code, which should be the case...

5.2 Functions in General: : — :, 10 min

- Computer programs can be very long and complicated
- Manage complexity by “boxing away” detail in functions
- A function is a program unit that can be called by name
- Contains program statements
- If a function calculates something, it may *return* it to the caller
- Avoids repetition if those statements will be called more than once
- Avoids bugs in that case
- Sure it will run same way each time with same inputs
- Also called “routine” or “procedure” or “subprogram”, w/ nuances
- Difference between “black box” and “glass brick”
- Programmer says name of function rather than inserting whole block of statements in the main routine
- $\sin(0.7)$, for example, has a numerical value
- Can put $\sin(0.7)$ anywhere a numerical value can go
- For example, you can say $x = \sin(\cos(x))$, just as in math notation
- Generally functions have few inputs and outputs

- Functions may have many internal variables not seen in main code
- Often write functions that will be called only once
- This simplifies code, like an outline

5.3 Using Git: : — :, 10 min

- Follow handout on a text file

5.4 Python Functions: : — :, 10 min

- Start with `def`
- Arguments in parens, comma-separated
- Statements are indented
- Have return values (even if `None`)
- Simple example:

```
def triple(x):  
    y = 3 * x  
    return y
```

```
triple(4)  
>>> 12
```

- Variables outside function can have different names from those inside:

```
z = 5  
triple(z)  
>>> 15
```

- It works, even though we used `z` and not `x`.
- Optional arguments have defaults assigned in definition
- To continue lines that don't have open parentheses/brackets use `\`
- `print_max.py` example
- Docstring
- Documentation with `python/module_template_with_docstring.py` (in `Files/python/`)
- `%doctest_mode` in `ipython`

5.5 Check In: : — :, 5 min

- Questions before we start?
- Do you have a terminal running python, and the lecture demo ready to go?
- Who has done the matplotlib tutorial? How much of pydatatut?
- Who has started reading a Python book?
- How is HW3? Quick look at it.
- “Resources” on homework are reading assignments and also suggestions for help sources.
- If you notice errors in equations, notes, solutions, etc., earn ~ 1 HW problem worth of extra credit for first time each reported.
- Bevington Eq. 4.3 on p. 52, Π should be π .
- Bevington Eq. 6.10 on p. 105, x should be x_i in first line and $\frac{1}{\sigma_i^2}$ should be $\frac{x_i}{\sigma_i^2}$ in last line.

5.6 Arguments in Python: : — :, 15 min

- Python has two types of arguments: positional and keyword
- Consider:

```
def triordup(x, y=None, prt1=False, prt2=False):
    z = 3 * x
    if prt1:
        print(x)
    if prt2:
        print(y)
    if y:
        z = 2 * y
    return z
```

- You can call this several ways (have students take turns predicting each outcome before executing):

```
triordup(3) # prints 9
triordup(x=3) # prints 9
triordup(3, 2) # prints 4
triordup(3, prt1=True) # prints 3...9
triordup(3, 2, prt2=True) # prints 2...4
triordup(3, 2, prt1=True) # prints 3...4
triordup(3, 2, prt2=True, prt1=True) # prints 3...2...4
```

- `x` is a positional parameter
- You know `x` is 3 and `y` is 2 from the position of `x` and `y` in the definition
- `prt1` and `prt2` have defaults,
- Keyword arguments can appear in any order
- Any argument can be omitted, but error if it doesn't have a default and gets used

5.7 Programming Style: : — :, 15 min

- `homework+coding.pdf` handout is in class handouts directory.
- Programming done in text.
- Text is communication; follow the 3 rules of communication:
 - Know your subject
 - Know your audience (viewer, listener, etc.)
 - Know exactly what you want your audience to know about your subject
- How do these apply to code?
 - Knowing subject can be hard! (algorithms and data can be complex)
 - Recipients: computer, you (sooner & later), colleagues, students (examples)
 - Must make all happy!
- Documentation with `module_template_with_docstring.py`
- Look at `goodcode.pro` and `badcode.pro` (using IDL code because IDL lets you not indent)
 - check input values
 - documentation
 - no gotos, no globals
 - indent for structure and readability
 - use horizontal and vertical spaces for clarity and ease of verification
 - Try not to hard-code values except 0 and 1! Put at top in variables if necessary, or pass into functions. (There are limits to this.)
 - Query data for items like image sizes that vary from one analysis to another, or pass them if necessary
- Document it if you modify an input (ok for arrays, otherwise avoid)

5.8 Astronomical Data: : — :, 25 min

- What is FITS?
- Demo: `05astropy.py`
- Reading FITS files
- Reading header variables
- Adding/modifying header variables
- Writing FITS files
- Display using DS9
- Example of `np.where()` and log scaling

Chapter 6

Measurement, Probability, and Statistics

6.1 Illegitimate, Systematic, and Random Errors: : — :, 10 min

- A measurement is the best estimate of a quantity in terms of accepted units.
- Three types of measurement error: illegitimate, systematic, and random
- Illegitimate error: a **fundamental** problem in the methods or assumptions (think you're doing one thing, really doing another)
- Find and fix!
 - Check yourselves! Most homework errors are in this category.
 - We're particularly picky in this course to prepare you for doing this in real science.
 - Mars Polar Lander English *vs.* Metric goof
- Systematic error: **predictable** difference between data and measurement
- Quantify and remove.
 - A non-quantified systematic error can kill a measurement
 - Must always estimate the unquantifiable error conservatively
 - Any experimental plan must show how it will quantify and remove systematic errors well enough to get the accuracy needed.
 - Usually this is a major portion of any proposal (lack of this is a “red flag” for a bad proposal)
- Random error: differences between data and measurement that can only be described **probabilistically**
- Quantify, propagate through equations/analysis
 - Called the “noise”

- Parameterized by the data's standard deviation
- Error bars in astronomy are 1σ (3σ in biology)
- Minimize by repeated measurement
- Figure of merit for all measurements: Signal-to-noise ratio (S/N or SNR or SN)

$$S/N = \frac{\text{final value of corrected measurement}}{\text{random error in that value}} \quad (6.1)$$

6.2 What's in a Measurement?: : — :, 10 min

- Attempt measurement of a person's height with tape measure bouncing on my shoe
- Have class call out all effects affecting the measurement and classify as illegitimate, systematic, or random
- What if stick was actually in half-inches, not centimeters, but we didn't notice?
- Non-trade scale demo

6.3 Terms: : — :, 10 min

- Language of probability is *precise*, due to the subtleties involved. Be careful!
- **accuracy**: closeness to truth
- **precision**: ability to make small distinctions
- (draw 2 charts like Bevington fig 1.1)
- **significant figures** (1000. vs. 1000)
- **error**: difference between measurement and truth (usually don't know)
- **estimate** (of error, mean, std. dev., etc.) best attempt to quantify (vs. true value)
- **uncertainty**: amount by which 2 measurements could differ and not be recognized as different
- **standard error**: estimate of 1σ (standard deviation) spread in measurements
- **significant difference**: different by several times the uncertainty (3 or preferably more)

6.4 Testing Theory: : — :, 10 min

An hypothesis or theory is clear, decisive, and positive, but it is believed by no one but the man who created it.

Experimental findings, on the other hand, are messy, inexact things which are believed by everyone except the man who did the work.

-- Harlow Shapley

- Good theories predict measurable quantities
- Do the measurement.
- If the measurements are within errors of the prediction, the data support the theory *at the level of their uncertainty*.
- Draw diagram w/ 2 curves that coincide for part of the span, and data in that region.
- Which is right?
- What to do next?
- Is it true?
 - If it conflicts with other theory, it must provide a replacement (or appeal to data that shows the other theory to be false).
 - It must not make predictions that are against the data, not even one.
 - Do other predictions also fit the data? Then need better data.
 - Do other theories that depend on this one also work?
 - Usually need to wait to see if other theories appear.
 - Eventually, theory becomes accepted if it passes these tests.
 - No theory is ever immune from being overturned later!

6.5 Probability Basics: : — :, 20 min

- Description of behavior of many identical, independently-prepared systems
- Discrete vs. continuous probability: coin vs. time interval
- In the limit of large N , everything appears continuous.
- Described by probability density function (PDF) that integrates to 1.

- Do definite integral over range to get probability:

$$P(x \text{ in range } a - b) = \int_a^b p(x)dx \quad (6.2)$$

$$\int_{-\infty}^{\infty} p(x)dx = 1 \quad (6.3)$$

- **mean:** $\bar{x} = \int_{-\infty}^{\infty} xp(x)dx$
- **median:** $P(x < \text{median}) = P(x > \text{median}) = 1/2$
- **mode:** $\max(p(x))$
- **variance:**

$$\sigma^2 = \int_{-\infty}^{\infty} (x - \bar{x})^2 p(x)dx \quad (6.4)$$

- mean is also the **expected value of** $x = \langle x \rangle$
- variance is the **expected value of** $(x - \mu)^2$, the square of the **deviations from the mean**
- **standard deviation:** σ
- **population:** all possible measurements of a system, proportionally represented (often infinite in number)
- **draw:** one measurement of system
- **sample:** a set of draws
- **estimate** mean, std. dev., etc. from sample

6.6 Estimates of Mean, etc.: : — :, 5 min

- All estimates use discrete formulae: can't take an infinite number of measurements!
- Estimate the mean and standard deviation of the parent distribution from a sample:
- **mean**

$$\bar{x} = \frac{1}{N} \sum x_i \quad (6.5)$$

- **standard deviation,** $\sigma = \sqrt{\text{var}(x_i)}$

$$\sigma^2 = \frac{1}{N - m} \sum (x_i - \bar{x})^2 \quad (6.6)$$

- Factor of $\frac{1}{N-m}$ takes into account effect of degrees of freedom, makes it work for low N .

Chapter 7

Probability Distributions and Error Analysis

Week 4.

7.1 Check In: : — :, 5 min

- Questions before we start?
- How was HW3?
- Comment on HW4:
 - A few months in the lab can save hours in the library!
 - Routines exist to do most of the hard work on HW4. Find them. Use them.
 - Part of the goal of the assignment is to get you to manipulate larger data entities than just single numbers. Be high-level, let the computer do the work.
 - The habit of testing your own answers is the highest goal of the class. Problem 2 tests problem 1, and item d of problem 3 is a test of the rest of the problem. Get in the habit. Always do it on your own data processing.
 - The assumption of others: you are wrong until you prove you are right.
- HW4 first “real” HW: exercises teach data analysis.

7.2 Gaussian Distribution: : — :, 10 min

•

$$p_G = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-x_0}{\sigma}\right)^2} \quad (7.1)$$

- Limit of sum of binomial (coin-toss) distribution (and most others) for large N

- “Normal” distribution - from psychology and biology trying to define what was a “normal” person, plant, animal
- $1\sigma = 68\%$ of measurements closer to mean than this
- $2\sigma = 95\%$ of measurements closer to mean than this
- $3\sigma = 99\%$ of measurements closer to mean than this
- “Well-behaved” errors follow it
- Memorize it!
- Not analytically integrable: Find function to calculate integral values.

7.3 Poisson Distribution: : — :, 15 min

- Timing of random, discrete events in time or space (diagram, don’t confuse x with space!)
- Example: Light!
- Say there are n events per unit time, on average
- Then $\tau = 1/n$ is the average interval between events
- In time t we would **expect** an **average** of $t/\tau = tn = N$ events
- The probability of getting x events in time t is

$$P(x, t) = \frac{(t/\tau)^x}{x!} e^{-t/\tau} = \frac{N^x}{x!} e^{-N} \quad (7.2)$$

- $\bar{x} = N$
- $\sigma = \sqrt{N}$
- For large N , Poisson approaches a Gaussian with $\bar{x} = N$ and $\sigma = \sqrt{N}$
- Gaussian, BUT mean and std. dev are locked together!
- \bar{x} is signal and σ is noise, so

$$S/N = \frac{\bar{x}}{\sigma} = \frac{N}{\sqrt{N}} = \sqrt{N} \quad (7.3)$$

- Quality of data improves as $\sqrt{\text{number of counts}}$...
- IF uncorrected systematics are well below σ .

7.4 Confidence Intervals: : — :, 5 min

- **Confidence interval:** probability that a draw lies inside a given range.
- For a Gaussian:
 - $1\sigma = 68\%$ of measurements closer to mean than this
 - $2\sigma = 95\%$ of measurements closer to mean than this
 - $3\sigma = 99\%$ of measurements closer to mean than this
- The 3σ confidence interval for a Gaussian is 99%.

7.5 Uncertainty of a Measurement: : — :, 5 min

- Estimate the measurement uncertainty by taking sample measurements, use σ
- Compare to *a priori* uncertainty estimate
- If rough agreement, use sample value, otherwise find problem

7.6 Goal of Error Analysis: : — :, 5 min

- How do you plug an uncertainty into a formula? Called error propagation.
- Given: variables: u, v, \dots
 mean values \bar{u}, \bar{v}, \dots calculated from samples with N_u, N_v, \dots draws u_i, v_i, \dots
 standard errors $\sigma_u, \sigma_v, \dots$
 function: $f(u, v, \dots)$
 want σ_f
- Calculate $f(\bar{u}, \bar{v}, \dots)$. This is the best estimate of \bar{f} .
- It is approximate if \bar{u}, \bar{v}, \dots calculated with small N , approaches true for large N .
- Plot a curve, choose x and σ_x , linearize at x , trace to f axis, show f and σ_f .

7.7 Error Propagation Equation: : — :, 15 min

- Consider the spread (variance) in values of $f_i = f(u_i, v_i, \dots)$:

$$\sigma_f^2 = \lim_{N \rightarrow \infty} \left[\frac{1}{N} \sum (f_i - \bar{f})^2 \right] \quad (7.4)$$

- We're ignoring the fact that it should be $N - m$ in the denominator, but m would be the number of variables in the equation.

- By the way, it is N in the denominator if the mean you are using is the real mean and not one estimated from the data. But, that rarely, if ever, happens.
- Also, if you get significantly different results with N or $N - m$, you're almost certainly up to no good and should get more data!
- First we need the deviations of f_i , which we can estimate with a Taylor series in the variables taken about their mean values:

$$f_i - \bar{f} \simeq (u_i - \bar{u}) \frac{\partial f}{\partial u} + (v_i - \bar{v}) \frac{\partial f}{\partial v} + \dots \quad (7.5)$$

- Combining these, we can get σ_f^2 in terms of $\sigma_u^2, \sigma_v^2, \dots$, which we measured:

$$\sigma_f^2 \simeq \lim_{N \rightarrow \infty} \frac{1}{N} \sum \left[(u_i - \bar{u}) \frac{\partial f}{\partial u} + (v_i - \bar{v}) \frac{\partial f}{\partial v} + \dots \right]^2 \quad (7.6)$$

$$\simeq \lim_{N \rightarrow \infty} \frac{1}{N} \sum \left[(u_i - \bar{u})^2 \left(\frac{\partial f}{\partial u} \right)^2 + (v_i - \bar{v})^2 \left(\frac{\partial f}{\partial v} \right)^2 + \dots \right] \quad (7.7)$$

$$2(u_i - \bar{u})(v_i - \bar{v}) \frac{\partial f}{\partial u} \frac{\partial f}{\partial v} + \dots \quad (7.8)$$

- We can distribute the outer sum over the terms.
- Then, since $\frac{\partial f}{\partial u}$, etc. are constants evaluated at \bar{f} , they come out of the sums and multiply something like Eq. 7.4.
- Substitute σ_u^2 , etc. for each occurrence of Eq. 7.4.
- Last term is the co-variance:

$$\sigma_{uv}^2 = \lim_{N \rightarrow \infty} \left[\frac{1}{N} \sum [(u_i - \bar{u})(v_i - \bar{v})] \right] \quad (7.9)$$

- So the **Error Propagation Equation** (memorize) is:

$$\sigma_f^2 \simeq \sigma_u^2 \left(\frac{\partial f}{\partial u} \right)^2 + \sigma_v^2 \left(\frac{\partial f}{\partial v} \right)^2 + \dots + 2\sigma_{uv}^2 \frac{\partial f}{\partial u} \frac{\partial f}{\partial v} + \dots \quad (7.10)$$

- If u and v are uncorrelated (one's not positive when the other is, and vice versa), then $(u_i - \bar{u})(v_i - \bar{v})$ will be negative half the time and positive half the time, and the sum will tend to zero.
- So, for uncorrelated variables, we can drop the co-variance terms.
- Equation is important for two reasons:
 - Evaluate σ_f
 - Determine which input variables have the greatest influence on the output, and therefore need the smallest errors.
 - Or on other words, how should I spend my telescope time?

7.8 HW3 Peer Evaluation: : — :, 60 min

- Use `plt.ion()` rather than `plt.show()`, or it will not run without human intervention.
- Use `print(x)`, not `print x` or just `x`.

7.9 Check In: : — :, 5 min

- Questions before we start?
- Homework must run or 10 points off
- If you are stuck, comment out stuff that doesn't run or use fake numbers to get it to run, and say so in a comment
- First quiz Thursday next week: Gaussian distribution, addition/subtraction problems w/ errors, error types

7.10 Error in Addition: : — :, 10 min

- This is important enough to memorize: spending extra class time on it!
- For addition, $f = u + v$, and all the partial derivatives are 1, so:

$$\sigma_f^2 \simeq \sigma_u^2 + \sigma_v^2 \dots (+2\sigma_{uv} \dots) \quad (7.11)$$

- “Add in quadrature”
- What is it for subtraction? Same, since the -1 in the partial derivative squares to 1.
- What if it's just $f = u + a$, with constant a ? Same as if $\sigma_v = 0$:

$$\sigma_f^2 \simeq \sigma_u^2 \quad (7.12)$$

$$\sigma_f \simeq \sigma_u \quad (7.13)$$

- What is $\begin{array}{r} 20 \pm 6 \\ + 60 \pm 8 \\ \hline \end{array}$

7.11 Error in Multiplication: : — :, 10 min

- For multiplication, $f = uv$:

$$\sigma_f^2 \simeq v^2 \sigma_u^2 + u^2 \sigma_v^2 \dots (+2uv \sigma_{uv}^2 \dots) \quad (7.14)$$

since $f = uv$,

$$\frac{\sigma_f^2}{f^2} \simeq \frac{\sigma_u^2}{u^2} + \frac{\sigma_v^2}{v^2} \dots (+2 \frac{\sigma_{uv}^2}{uv} \dots) \quad (7.15)$$

- How about multiplication by a constant, $f = au$? Then $v = a$ and $\sigma_v = 0$

$$\sigma_f^2 \simeq a^2 \sigma_u^2 \quad (7.16)$$

$$\sigma_f \simeq a \sigma_u \quad (7.17)$$

- How does this apply to powers (**careful!**)?
- How about division, $f = au/v$? Then $v' = v^{-1}$, and there's a -1 in front of $\frac{\partial f}{\partial v}$. It gets squared to 1 in the straight term but not in the cross term:

$$\frac{\sigma_f^2}{f^2} \simeq \frac{\sigma_u^2}{u^2} + \frac{\sigma_v^2}{v^2} \dots (-2 \frac{\sigma_{uv}^2}{uv} \dots) \quad (7.18)$$

- What is 16 ± 8
 $\times 32 \pm 8$

7.12 Error in Mean and Wrapup: : — :, 10 min

- How about the error in the mean of many measurements?

$$\bar{x} = \frac{1}{N} \sum x_i \quad (7.19)$$

That's a sum and a multiplication. If $\sigma_{x_i} = \sigma_x$, then

$$\sigma_{\bar{x}}^2 \simeq \left(\frac{1}{N}\right)^2 N \sigma_x^2 = \frac{\sigma_x^2}{N} \quad (7.20)$$

$$\sigma_{\bar{x}} \simeq \frac{\sigma_x}{\sqrt{N}} \quad (7.21)$$

- The error in the mean improves with \sqrt{N} .
- NOTE: This is not the scatter in the measurements, it is the error in the mean estimated from the measurements.

- Remember: errors add in quadrature, sometimes with a weighting.
- It's easiest to work with variances rather than uncertainties.
- Recall that in the Poisson distribution, the variance is the value!
- Chapters 3 and 4 show how to average according to weights calculated from uncertainties.
- They also discuss pragmatic reasons for doing this *vs.* discarding old data, and the limitations to taking more data to improve σ through large N .

7.13 HW4 Peer Evaluation: : — :, 50 min

- Please do not quote the HW problem text. In sequential problems, it makes reading your answers difficult.
- Non-ASCII characters like the quotation marks and times symbol (gotten from PDF cut-and-paste) prevent Python from running.
- Do not hard-code the directory pathname for writing, or we will not be able to run your code.

7.14 Computational Error Analysis (Monte Carlo): : — :, 5 min

- MODERN LIFE IS EASIER:
- Program the formula into a computer
- Generate samples of fake measurements distributed according their uncertainties.
- Evaluate the function for each draw in the sample.
- Calculate the uncertainty.
- Note: This gives an answer, but no information on which parameters it's sensitive to.
- To do that, vary each σ_i and do the above many times.
- Look at how σ_f changes with changes in each σ_i .

Chapter 8

Signal-to-Noise Ratio Calculations

8.1 Calculating Astronomical Signal-to-Noise Ratios: : — :, 5 min

- The idea is simple...
- $\text{SNR} = S/N = \frac{\text{value}}{\text{uncertainty}}$
- You already know how to calculate uncertainties, numbers of photons, noise from photon statistics and from detectors
- But, it can get complicated, even if it's "just" a lot of simple things contributing
- Number of photons from anything: $E/h\nu = E\lambda/hc$
- Intrinsic noise in any measurement based on photons (Poisson noise): $\sigma_N = \sqrt{N}$
- The ultimate S/N limit: $S/N = N/\sqrt{N} = \sqrt{N}$.
- Remember, N is total photons, not photon count rate (N/t), not energy, not counts (DN)!

8.2 Simple Scaling Rules: : — :, 5 min

- The more you look, the better S/N gets, by \sqrt{N}
- N goes up linearly with time (usually), so S/N goes up as \sqrt{t}
- N goes up with *area* of telescope, so S/N goes up *linearly* with telescope diameter
- A 10 meter telescope collects $100\times$ the photons of a 1 meter, but has only $10\times$ as good S/N
- N goes up with $\Delta\lambda$, but not necessarily linearly
- N goes down with distance d^2 , so S/N drops as $1/d$

8.3 Calculating Signal: : — :, 5 min

- Usually the signal is a single item (just a planet, just a star, the value of an object's spectrum in a single wavelength channel)
- Calculate the number of photons or electrons

$$N = \frac{At}{4\pi d^2} \int_{\Delta\lambda} F_{\text{obj}}(\lambda) T(\lambda) d\lambda$$

- Flux F_{obj} is measured at object
- For a spherical blackbody (ok approximation for a star or planet), $F = 4\pi R^2 \sigma T^4$
- $A/4\pi d^2$ is fraction of that entering telescope
- T is optical transmission

8.4 Calculating Noise: : — :, 10 min

- All the noise from all sources counts - add them up
- Variances add, so work in variances of photons or electrons
- For Poisson sources, variance is N
- For noise sources, variance is σ^2
- If you subtract an image, add its variance
- If you sum a region (aperture or annulus), it's an area, so sum the area times the variance per pixel
- If you multiply an area by a rate, sum SQUARE of area by variance! (Why?)

8.5 Signature of Unknown Errors: : — :, 5 min

- Signal-to-noise ratio ($S/N = \frac{\text{\# photons from object}}{\text{noise from all sources}}$)
- For pure Poisson noise (e.g., photon noise from a star), $S/N = \sqrt{\text{\# of photons}} = \frac{N_*}{\sqrt{N_*}}$
- N on left is noise, on right is number of photons from star.
- Generally, denominator will have many terms, including read noise, noise from sky variability, and systematic errors.
- What if S/N stops improving as you observe more?

- Then, you have some noise source that is dominating the photon noise.
- You need to identify this and either correct it in the observing system or correct it in the analysis.
- It is not always possible to make such corrections!
- Common problems include sensitivity variations with location of stellar center in a pixel (e.g., Spitzer InfraRed Array Camera channels 1 and 2 at 3.6 and 4.5 μm).

8.6 Practice: Exoplanet S/N: : — :, 10 min

- Need to add errors for signal-to-noise-ratio (S/N) calculations
- Consider exoplanet secondary eclipses
- There is a star of flux F_*
- There is a planet of flux F_p
- The signal is F_p
- The noise source is all the photons, both star and planet
- $\sigma = \sqrt{\sigma_*^2 + \sigma_p^2}$
- The inherent S/N limit is:
- $S/N = F_p / \sqrt{\sigma_*^2 + \sigma_p^2}$
- It's actually a bit worse
- We measure F_{*+p} for some amount of time before and after eclipse
- We measure F_* alone during eclipse for a different amount of time
- We calculate $F_p/F_* = (F_{*+p} - F_*)/F_* = F_{*+p}/F_* - 1$
- The flux uncertainties depend on the observing time for each flux
- We must also add in noise from the detector (read noise, dark current noise, flat field noise, sky noise, etc.)
- Remember that noises add in quadrature
- Remember to consider each detector read, all seconds of integration, etc.
- Fluxes can be subtracted, but error always adds!

8.7 Quiz 1: : — :, 30 min

Chapter 9

Fitting

9.1 Check In: : — :, 10 min

- Questions before we start?
- First quiz Thursday: Gaussian distribution, addition/subtraction w/ errors, error types
- What is probability? How do we quantify it?
- What is the mean? standard deviation? mode? median?
- What is Poisson distribution? Gaussian?
- Describe the statistics of the arrival of photons.

9.2 Fitting Basics: : — :, 10 min

- Have a set of theoretical equations, or a **model**.
- Some parameters are known, some are not. Unknowns are called **free parameters**.
- Want to find values of the free parameters so the model fits the data as well as possible.
- **MUST** have more data points than free parameters!
- Can estimate both parameters and their uncertainties, if enough data.
- Generally use maximum likelihood to decide which model is best (least squares is a special case).
- Sometimes can be done with minimization calculus, other times must be done numerically (see Bevington Ch. 8).
- There are Python routines for line fits and for generalized fits.
- Often problems can be reduced to or approximated by line fits.
- Always check goodness of fit, not just errors!

9.3 Least Squares Linear Fitting: : — :, 25 min

- Model: $y = a + bx$, measurements (x_i, y_i) . Assume uncertainty σ_i , mainly in y (or flip).
- Want to minimize $\Delta y_i = y_i - y(x_i) = y_i - a - bx_i$
- Assume Gaussian-distributed measurements. Then the probability of getting any single measurement is:

$$P_i = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp \left\{ -\frac{1}{2} \left[\frac{y_i - a - bx_i}{\sigma_i} \right]^2 \right\} \quad (9.1)$$

- Probability for the whole dataset is:

$$P(a, b) = \prod P_i = \prod \left(\frac{1}{\sqrt{2\pi\sigma_i^2}} \right) \exp \left\{ -\frac{1}{2} \sum \left[\frac{y_i - a - bx_i}{\sigma_i} \right]^2 \right\} \quad (9.2)$$

- We want to maximize $P(a, b)$. a, b appear in the exponent, which is negated, so minimize it:

$$\chi^2 = \sum \left[\frac{1}{\sigma_i} (y_i - a - bx_i) \right]^2 \quad \text{least squares} \quad (9.3)$$

$$\frac{\partial}{\partial a} \chi^2 = \frac{\partial}{\partial a} \sum \left[\frac{1}{\sigma_i} (y_i - a - bx_i) \right]^2 \quad (9.4)$$

$$= -2 \sum \left[\frac{1}{\sigma_i^2} (y_i - a - bx_i) \right] = 0 \quad (9.5)$$

$$\frac{\partial}{\partial b} \chi^2 = \frac{\partial}{\partial b} \sum \left[\frac{1}{\sigma_i} (y_i - a - bx_i) \right]^2 \quad (9.6)$$

$$= -2 \sum \left[\frac{x_i}{\sigma_i^2} (y_i - a - bx_i) \right] = 0 \quad (9.7)$$

$$(9.8)$$

- Rearrange:

$$\sum \frac{y_i}{\sigma_i^2} = a \sum \frac{1}{\sigma_i^2} + b \sum \frac{x_i}{\sigma_i^2} \quad (9.9)$$

$$\sum \frac{x_i y_i}{\sigma_i^2} = a \sum \frac{x_i}{\sigma_i^2} + b \sum \frac{x_i^2}{\sigma_i^2} \quad (9.10)$$

$$\text{OR:} \quad C = aD + bE \quad (9.11)$$

$$F = aE + bG \quad (9.12)$$

- Solve to get:

$$a = \frac{1}{\Delta} (GC - EF) \quad (9.13)$$

$$b = \frac{1}{\Delta} (DF - EC) \quad (9.14)$$

$$\Delta = DG - E^2 \quad (9.15)$$

9.4 Function Minimization and Goodness of Fit: : — :, 10+5 min

- Linear fitting doesn't work if there is more than a multiply and an add parameter
- E.g., multiply, add, and shift
- Could derive a new expression like linear least squares formulae
- Or, do it numerically:
 - Define a Python function that has parameters of model as inputs
 - Usually the parameters are in a vector
 - Run `scipy.optimize.leastsq` on the data with the function
 - It finds the optimal values, and errors!
 - Can be slow, depending on model and data space
 - Can be fooled if data space is strange: local minima in χ^2 space
- What `scipy.optimize.leastsq` does is:
 - Evaluate the function
 - Compute χ^2 vs. data
 - Vary the parameters and repeat
- It explores parameter space, looking for the minimum χ^2
- It uses a function minimization routine
- Calculates gradients in χ^2 and walks down them
- Parameter errors are related to second derivatives near χ^2 minimum
- `06fitting`

9.5 HW5 Peer Evaluation: : — :, 60 min

Chapter 10

Astronomical Measurements, CCD Arrays, Array Correction

10.1 Check In: : — :, 5 min

- Questions before we start?

10.2 Types of Astronomical Measurements: : — :, 5 min

- Imaging: positions
- Photometry: brightnesses
- Spectroscopy: wavelength dependence of light
- Often combine 2 of these: hyperspectral imaging, spectrophotometry, imaging photometry
- Variation in time, often at high cadence

10.3 What's After the Telescope?: : — :, 15 min

- Telescope gathers light falling on many square meters
- Makes image in its focal plane (draw with focus way back to leave room for elements)
- Signal-sorting elements
 - Filters (show response function)
 - Dispersers (prism, grating, grism)
 - Reimaging optics
 - Adaptive optics
- Detecting the final signal


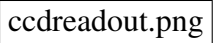
- QE = quantum efficiency, fraction of photons detected
- Eye/sketchpad — ok for imaging, sensitivity bad and qualitative, cats QE 10%, human few %
- Photographic film — ok for imaging, sensitivity poor, but quantitative, QE 10%
- Rapid sequence of detectors, like vidicons, electronic cameras
- Photomultipliers — 1D photometry only, accurate, fast, QE 50+%
- Array detectors — now almost all detectors except at radio and γ -ray wavelengths, QE 80-95%

10.4 Array Basics: : — :, 10 min


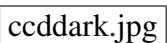
- Semiconductor chips sensitive to light
- 2D array of “wells” (they produce pixels, and are sometimes even called pixels)
- Materials used determine wavelength response
- 2 types: CCD and CMOS array, for visible and IR (CMOS becoming popular in digicams)
- QE 80-95%, near theoretical max.
- Quantitative, dimensionally near perfect
- Formats vary, good optical CCD now 4k×4k
- Mosaics up to 38k×38k = 1.4 billion pixels per image! (PAN-STARRS)
- Big dynamic range, 12–24 meaningful bits common, 2-4 bytes/pixel
- Monochromatic: must take 3 frames with different filters for true color
- Pixel size 10–30 μm
- There is a supplemental reading in `site-ccd.pdf` and `CCD.Manual.2010.pdf` for those interested in more about CCD chips and CCD observing.
- Read PAN-STARRS web page on camera itself: pan-starrs.ifa.hawaii.edu

10.5 From Photon to Data: : — :, 20 min

- `ccdwell.png`
- Photon penetrates into Si, is absorbed
- Energy knocks e^- into conduction band
- Every third well is biased (+10 V) for integration

- e^- attracted to biased well, is trapped
- Pixel integrates photoelectrons
- 
- To read out, adjacent gate also biased, well is wider
- Original gate bias reduced, e^- flow to new well
- Original gate bias zero, e^- have moved
- Repeat 3 times to move entire image 1 pixel
- 
- Last row is called shift register
- It shifts the same way
- The last pixel in the shift register reads into an amplifier, then ADC
- Then all pixels in shift register move over 1
- Repeat until shift register empty
- Clock another row
- Whole array reads in few sec, or fraction of sec
- Readout times depend on size and clocking rates chosen
- Faster clocking → more noise in amplifier
- Slower clocking → get closer to saturating

10.6 CCD Characteristics and Systematics: : — :, 15 min

- 
- Quantum efficiency mainly in visible, cut off at 1.1 μm .
- Modern QE $\sim 80+\%$
- QE varies from pixel to pixel by 10% or more
- 
- Dark Current: thermal vibrations in Si can bump e^- into conduction band

- Charge integrates even in dark!

$$I_d \sim \exp(-a/kT) \quad (10.1)$$

- Strongly temperature-dependent, so CCDs usually cooled by LN_2 to ~ 77 K
- On-chip amplifier has a bias voltage to attract the charge from the shift register
- It also has a gain:

$$V_{\text{out}} = gN_{e^-} + V_{\text{bias}} \quad (10.2)$$

- V_{out} digitized into ADU (Analog-to-Digital Units) or DN (Data Numbers)
- Amplifiers are not perfect, pixel capacitance adds “read noise”
- Good CCDs have $< 5 e^-$ read noise, or $< 25 e^-$ charge added per pixel

10.7 Check In: : — :, 5 min

- Questions before we start?

10.8 Summary of CCD Systematics: : — :, 5 min

- QE $< 100\%$
- QE varies
- I_d
- I_d varies pixel to pixel
- I_d varies in time (with temp, etc.)
- I_d has its own noise
- V_{bias}
- Read noise
- Amplifier not perfectly linear
- Pixels not perfectly linear! Too much charge cancels some bias voltage.
- Cosmic rays deposit many e^- in wells
- Some pixels have defects (“bad pixels”)

10.9 Basic CCD Error Correction: : — :, 5 min

- Do readout backwards:
- QE differences were multiplied, so divide them out
- Dark current was added, so subtract it
- Bias voltage was added, so subtract it
- Basic formula:

$$C = \frac{R - B - D}{F} \quad (10.3)$$

- `ds9 ori.fits`, show bleeding, bias pattern, hot/cold columns, etc.
- Q: What would a bad column in every frame going from some spot to the top mean?
- Q: If each frame were bad to the left of some constant point, what would that mean?

10.10 Making the Calibration Frames: : — :, 10 min

- This is an art!
- Sensitive analyses and quirky instruments → more robust methods
- Here are the basic principles
- Bias frame: average many 0-exposure images, $B = \frac{1}{N} \sum B_i$
- Dark frame: $D = \frac{1}{t_{\text{tot}}} \sum (D_i - B) = \frac{1}{t_{\text{tot}}} \sum (D_i) - \frac{NB}{t_{\text{tot}}}$
 - average many unlit images with your exposure time
 - ...or, take long exposures, subtract mean bias, sum, divide by total itime
 - ...or fit line to image values for each pixel position, finding bias and dark rate
 - Can then reconstruct dark current for range of exposure times
 - Beware of big extrapolations: best to avoid if possible
 - Dark current is usually very small in the visible

- Flat field:

$$F = \frac{\sum (F_i - B - D)}{\langle \sum F_i - B - D \rangle_{\text{all}}} \quad (10.4)$$

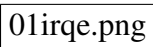
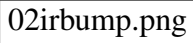
- Take many images of uniformly-lit, defocussed subject, half-well
- Subtract dark and bias
- Average
- Normalize so mean value ~ 1

- Remaining instrument systematics are consistency problems
- For consistency problems, take flats/biases through night and use only those nearest in time to exposure.
- Need flats in each filter!

Chapter 11

Infrared Arrays, Sky, and Corrections

11.1 Infrared Arrays: : — :, 20 min

- Can dope Si so it is sensitive in visible and still act as CCD
- Need other materials to be sensitive at long wavelengths
 - InSb $\rightarrow \lambda < 5 \mu\text{m}$
 - HgCdTe $\rightarrow \lambda < 2.4 \mu\text{m}$
 - HgCdTe $\rightarrow \lambda < 10 \mu\text{m}$
 - Si:As $\rightarrow \lambda < 28 \mu\text{m}$
 - Si:Sb $\rightarrow \lambda < 40 \mu\text{m}$
 - HgCdTe QE curve 
- Most IR materials can't work as semiconductor like Si CCDs
- Also, the background can be high in the IR
- Air, telescope, dome, and camera can emit in IR brighter than astronomical sources
- “Background limited” sources fill wells fast, so need to read out up to 1 kHz
- CCD takes several seconds to read
- Solution: IR absorbing layer connected to Si multiplexer
- Indium bump bonds connect them 
- Each unit cell reads out directly in turn to amplifier: no charge motion across chip
- Often have many amplifiers, each reading quadrants or every 4th row, etc.
- Huge read noise, up to $2000 e^-$
- Can measure voltage on each unit cell diode more than once: non-destructive reads (NDR)

- Correlated multiple sampling reduces read noise by $\sqrt{\text{NDR}}$
- Trade-off between read speed and noise: CCD slow and accurate, multiplexer fast and noisy
- Major problems w/ thermal contraction mismatch, can crack

11.2 Characteristics of Infrared Arrays: : — : , 5 min

- More susceptible to dark current
- Also, camera has actual thermal emission
- So, cool to 4 K w/ IHe
- QE \sim same as CCDs
- QE varies more from pixel to pixel
- More bad pixels/regions than CCDs (bumps separate, cracks, non-Si technology lags)
- Same additive and multiplicative effects in multiplexer as with CCD, but worse
- Systematics have much more drift during night
- Also need to correct background (sky+telescope+camera, called sky)
- IR array correction formula:

$$C = \frac{R - S}{F} \quad (11.1)$$

- [08irarraybasics/irarray_correction.pdf](#)
- S is sky image taken close to exposure, includes bias, dark current, telescope, camera, etc.
- Different techniques for getting flats and sky, depending on λ

11.3 Near-IR Sky: : — :, 10 min

- Sky is dark in the visible, except scattered moonlight and light pollution
- Usually estimate sky brightness from nearby pixels in visible (plot stellar PSF cross-section)
- In infrared, atmospheric gases emit and absorb radiation
- Near-IR bands [03-06skytrans](#):
 - J: 1.1–1.4 μm
 - H: 1.5–1.8 μm
 - K: 1.95–2.4 μm

- Emission from excited OH⁻ in atmosphere 07-09ohsky_jhk
- Varies on timescales of several minutes
- Nod/dither to get sky frame
- Usually good to interpolate nodded sky images in time
- Data and sky exposures must fit into few minutes, taken in pairs or groups

11.4 Mid-IR Sky: : — :, 10 min

- Mid-IR: Where the sky glows
- Significant thermal emission from sky at $\lambda > 4 \mu\text{m}$
- At $5 \mu\text{m}$, sky much brighter than Jupiter
- Varies on timescales of 1 sec or less
- Often chop secondary mirror in mid IR at $\sim 5\text{--}10 \text{ Hz}$ (for $\lambda > 10 \mu\text{m}$)
- Decision for when to switch from nothing to nodding to chopping is site-dependent
- At $>5 \mu\text{m}$, you can observe in the day! Sun dim and sky bright.

11.5 Array Effects: : — :, 10 min

- Summary of array effects
 - Sky background (+, smooth, time-variable)
 - Dark current (+, smooth and not, mostly steady)
 - Biases from multiple amplifiers (+, not smooth, steady)
 - QE (\times , not smooth, steady)
 - Cosmic rays (X, not smooth, time-variable)
 - Bad pixels (X, not smooth, steady)

- High on board:

$$ADU = g_2(g_1(E_Q(F_* + F_{\text{sky}}) + I_d) + V_{\text{off}}) \quad (11.2)$$

$$ADU = g_2g_1E_QF_* + g_2g_1E_QF_{\text{sky}} + g_2g_1I_d + g_2V_{\text{off}} \quad (11.3)$$

- g_1 Volt/electron from unit cell
- g_2 ADU/Volt from array amp + ADC
- E_Q quantum efficiency

- F light fluxes
- I_d dark current
- V_{off} bias voltage on unit cell gate
- Single image: all array effects present, many vary spatially, often w/same appearance as real objects
- Stars, galaxies, planets also vary spatially
- Must move completely off objects for sky frame to disentangle signal from noise
- If sources take much of frame, nod off frame
- If sources small, nod can be small and objects can still be on frame: dithering (jittering in spectroscopy)
- `ds9 09medcomb/stars*raw*`, color, frame/match colorbars, frame/blink
- No time “wasted” getting sky since all frames have objects `09medcomb/10histogram.jpg`
- Stars/galaxies/etc. shift to different pixels, steady
- Noise/systematics stay in place, may average out in time

11.6 HW6 Peer Evaluation: : — :, 60 min

Chapter 12

Combining Frames

12.1 Check In: : — :, 5 min

- Questions before we start?
- How was HW5?
- DPS

12.2 Flats: : — :, 5 min

- Each flux value still multiplied by $g_1 g_2 E_Q$
- 1 million unknowns in image! Want to remove.
- Take image of uniform source, remove array effects, divide by median.
- Still don't know g_1 , g_2 , or E_Q , but know $k g_1 g_2 E_Q$ product at each pixel!
- Dividing by this corrects to $k F_*$.
- Calibrate k by observing standard stars.

12.3 Median Combination: : — :, 40 min

- [09medcomb/normalized_median_combination.pdf](#)
- Can use median combination to make flats, darks, biases, skies
- Sigma-rejection technique from HW5 is even better, given enough images
- If sky varies *and* you are making skies, add indented *normalized* median combination steps
- Take many dithered images

- Subtract dark frame (cold shutter, “closed” filter)
- Divide each frame by its median (median is now 1, may use median region rather than full frame)
- Loop over pixel positions
- Find median of all images at each pixel position
- Astronomical sources, cosmic rays, bad pixels rejected! Noise reduced!
- Save as normalized median
- Subtract normalized median from dark-subtracted, normalized images
- Multiply each resulting image by the median it was divided by
- Demo: `09medcomb/medcomb-demo`

12.4 Cleaning and Producing Sky Flats: : — :, 10 min

- I_d significant, as are thermal effects
- Flats are images of uniform sources of light
- Must subtract low-light and high-light w/ same itime
- Screen with lamp on/off
- Take many exposures (cosmic rays, build S/N)
- Take many exposures with lamp off, same itime
- Median combine lamp-on and lamp-off frames separately
- Subtract the lamp-off median frame from the lamp-on median frame
- Divide by median of middle area so values ~ 1
- `dark, lamp, flat`
- Sky (except stars) in early dawn may be “more flat”
- But, need just right light levels: planning!
- Park telescope so stars whiz through and don’t burn in
- Take many exposures at two different light levels
- Median-combine, subtract, and divide as above
- Star streaks should disappear in median combination (or too few frames)

12.5 Coding an Analysis: : — :, 10 min

- Data are sacred, even directory times! Never run in a data directory.
- Begin with the end in mind and plan:
 - What plots will I want in my paper? What tables? What images?
 - Plan the major steps and intermediate data products to get them.
 - Figure out what calculations you will do (these become functions).
- Use top-level files of commands specific to analysis
- Write general routines called by those specific files
- Put saves after big CPU waits, then break files after saves
- Pass or compute values like sizes that vary from one analysis to another

12.6 Software Implementation: : — :, 10 min

- Good data reduction pipelines follow homework format
- Look at HW solutions to see how to break out values from code
- Dated log file has notes and commands tested (not distributed)
- File(s) of naked commands or high-level functions
 - Have things specific to data
 - Define order of what happens to data
 - Can be batch-run as Unix command
- General routines get called from batch file
- Use strings to list files, do not iterate over numbers and insert into rootname unless certain there are no bad frames
- Allows you to eliminate bad files in list easily
- Separate extension and directory makes it easy to insert substrings in output filenames
- Query string arrays for number of files (for array preallocation) w/ `.shape`
- Can use optional keyword parameters (`foo=value`) and set defaults:

```
def foo(x, bar=2):
    return x * bar
```

...or...

```
def foo(x, verbose=None):
    if verbose is not None:
        print("Verbose was set! Say something!")
        bar = 3
    else:
        bar = 2
    return x * bar
```

- Modifying passed data is bad unless this is the main point of the routine. Document loudly!
- Work on a copy instead: `xc = x.copy`

12.7 Quiz 2: : — :, 20 min

12.8 FITS Implementation: : — :, 20 min

- FITS = Flexible Image Transport System, a data file format
- Developed in 1970s, among oldest data formats
- FITS header is set of 80-column “cards” (throwback!)
- Data in IEEE binary format
- Can also do ASCII tables, repeated header/data units, etc.
- FITS mainly used for images with 1 HDU = header-data unit = 1 header + 1 dataset
- Header documents origin of image, what has been done to it
- FAST: `less SPITZER_I4_15891712_0692_0000_4_bcd.fits`
- Point out keyword, value, comment, HISTORY, END, NAXIS array
- Related items can be in 1D array, show NAXIS*
- Analysis may compute many auxiliary values per image
- If distributing (“transporting”) data to others
 - Can store auxiliary values in header to inform recipient of...anything
 - Make HISTORY entries to summarize your processing and new values
 - `less rdpharo.py` to see HISTORY being made!
 - If adding several keywords, pick related names (NORMX1, NORMY1, not X1, Y2)

- PyFITS produces arrays of HDU objects, and methods to operate on them
- `fitsdemo.py`, pointing out:
 - `import astropy.io.fits`
 - `data, hdr = astropy.io.fits.getdata('file.fits', header=True)`
 - Data object is an ndarray
 - Header is a new object, address either as list or dictionary(!):
 - `print(hdr['NAXIS'])` Index header like dictionary
 - `print(hdr[3])` Index header like list (or tuple or array of strings)
 - `hdr['filter'] = 'FW555'` Assign to existing variable
 - `hdr['flatfile'] = 'flat17.fits'` For a completely new variable
 - FIRST header entry matching a keyword applies, in case of repeated entries, except COMMENT and HISTORY
 - See demo, astropy help, or Python Data Tutorial for reading and writing keywords and history (but note updates since then).
 - Use `astropy.io.fits.getdata()`, `astropy.io.fits.getheader()`, `astropy.io.fits.writeto()` for FITS file I/O (others available)

Chapter 13

PSF and Photometry

13.1 Check In: : — :, 10 min

- Questions before we start?
- What are the most basic differences between CCDs and IR arrays?
- What's the basic idea behind median combination?
- What are some circumstances when median combination doesn't work?
- Halfway through course, starting the first data analysis. Reread coding handout online. Review coding notes.
- How was HW6? (Fingers up for number of hours spent, record estimate.)
- Error in Howell: p. 105, it's a $(2L + 1) \times (2L + 1)$ box, not a $2L + 1 \times 2L + 1$ box.

13.2 N -Dimensional Gaussian: : — :, 10 min

- We often use a 2D Gaussian to model a stellar PSF
- Usually symmetric, needn't be
- Consider

$$p_G = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-x_0}{\sigma}\right)^2} \quad (13.1)$$

- Integrates to 1. For 2D function, y direction must integrate to $p_G(x)$ at each x .
- Since p_G integrates to 1, just multiply: $p_G(x)p_G(y)$
- This smears x Gaussian in y .

- Repeat for higher dimensions

$$p_{G,N} = \prod_{i=1}^N \left(\frac{1}{\sqrt{2\pi\sigma_i^2}} \right) e^{-\frac{1}{2} \sum_{i=1}^N \left(\frac{x_i - x_{i,0}}{\sigma_i} \right)^2} \quad (13.2)$$

- Note exponent now looks like an ellipse, hence name elliptical Gaussian.
- See `gaussian.py` in class Python dir.

13.3 Convolution: : — :, 10 min

- Convolution describes signal spreading (among other things)
- PSF
- Divide signal into discrete points
- Multiply each point by a function
- Replace point with function, shifted to location of point

$$(f * g)(t) \equiv \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau = \int_{-\infty}^{\infty} f(t - \tau)g(\tau)d\tau \quad (13.3)$$

$$(f * g)(n) \equiv \sum_{m=-\infty}^{\infty} f(m)g(n - m) = \sum_{m=-\infty}^{\infty} f(n - m)g(m) \quad (13.4)$$

- Demo movies
- In astronomy, kernel often symmetric
- Used for smoothing
- Can be calculated fast with Fourier Transforms, but don't, use `np.convolve()`

13.4 HW7 Peer Evaluation: : — :, 60 min

13.5 Point-Spread Function (PSF): : — :, 20 min

- Consider a perfect, point-like light source, far away
- Atmosphere has cells of air w/ differing temperatures
- Cells refract light, also move

- High-speed images show stellar point source bouncing around
- [11PSF/Eps_aql_movie_not_2000.gif](#)
- Central limit theorem → Gaussian
- Airy diffraction pattern (see next section)
- Telescope aberrations (astigmatism, coma, etc.)
- All these effects produce PSF
- Convolve “reality” image with PSF → observed
- Demo: [11PSF](#), cuts in X and Y, see steps in `psfdemo.py`
- Why are bright stars bigger? Same PSF!
- In the atmosphere and for long space exposures, every image, and sometimes different parts of the same image, have a different PSF

13.6 Airy diffraction Pattern: : — :, 10 min

- The Fraunhofer diffraction pattern [11PSF/Airy-pattern.png](#)
- ...is described by the Airy function:

$$I(\theta) = I_0 \left[\frac{2J_1(x)}{x} \right]^2 \quad (13.5)$$

$$x = ka \sin \theta = \frac{2\pi a}{\lambda} \frac{q}{R} = \frac{\pi q}{\lambda N} \quad (13.6)$$

$$I_0 = \frac{P_0 A}{\lambda^2 R^2} \quad (13.7)$$

where

I is the intensity,

θ is the angle from the optical axis,

I_0 is the intensity in the center,

$J_1(x)$ is the first-order Bessel function of the first kind,

$k = 2\pi/\lambda$ is the wavenumber,

a is the aperture radius,

λ is the observation wavelength,

q is the distance from the optical axis,

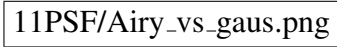
R is the distance from the aperture center ($q/R = \sin(\theta)$),

$N = R/2a$ is the $f/\#$, and

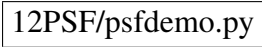
P_0 is the total power illuminating the aperture from the point source.

- Radius of first minimum (Airy disk radius):

$$r_D = \frac{1.22\lambda}{d} \quad (13.8)$$

- The maximum intensity of the first ring is about 1.75% of the central peak intensity
- Very close to a Gaussian 

13.7 Centering: : — :, 5 min

- Need to know exact (sub-pixel) center of PSF for photometry
- Many methods
- For most stars, fit a 2D Gaussian with free x and y parameters
- For Spitzer IRAC data, good to 0.02 pixels!
- See gaussian.py in class directory
- For very dim stars, noncircular PSFs, and non-stars, center-of-light (Howell p. 105) may be better
- See Files/python/gaussian.py
- 

Chapter 14

Aperture Photometry

14.1 Data Formation Model (“A Star is Born...In Python”): : — :, 5 min

- [13apphot/apphot-demo.pdf](#)
- Start by making a fake image of a star, with simulated photon noise
- First the background level
- Then its noise
- Then their sum
- Then the ideal star PSF
- Then its noise
- Then their sum
- Finally, the sum of background and star images
- This is our model of how the image was formed

14.2 Choosing the Star Aperture and Sky Annulus: : — :, 5 min

- We want to estimate the flux of the star
- Some sky remains (even with background image subtraction)
- How far out from star center to count?
- Establish center accurately (see Centering)

- Look at radial profile of star
- Want all of starlight, but minimal sky noise
- How far out to count? (see radial profile)
- The optimal aperture size for best S/N varies with S/N (see demo)
- Must do same apertures for all stars to avoid systematic error
- Often, we compare photometry done with different apertures
- Check your aperture size!
- Establish σ using bright star (why bright?)
- Use `fitgaussian` or just do FWHM and convert
- Count DN within $3-4\sigma$ from center for star
- Count DN in sky annulus farther out
- Want many more sky pixels than aperture pixels, for better S/N

14.3 Aperture Photometry Formula: : — :, 5 min

- Define an aperture
- Sum the flux in it
- Count the pixels in it
- Define an annulus away from the star but centered on it
- Sum the flux in it
- Count the pixels in it
- Do not count bad pixels in sky region (why must fix and count for star?)
- Table:

	Light (DN)	Area (pix ²)	Bad Pixels
Star	F_*	N_*	must fix
Sky	F_{sky}	N_{sky}	don't count
- Aperture photometry equation:

$$F_{*,\text{cal}} = F_{*,\text{raw}} - N_* \frac{F_{\text{sky}}}{N_{\text{sky}}} \quad (14.1)$$

- Or, subtract mean sky from image, then sum aperture (same thing)

14.4 Apertures in Python: : — :, 10 min

- Make and multiply masks: 3M method!
- Cut out area with star to 10σ or more
- Find PSF center
- Make mask of aperture and annulus, same size as cut-out
- Multiply by image
- Sum masked images for fluxes, sum masks for areas (numbers of pixels)
- To make annulus mask, make 2 concentric disk masks and subtract
- Multiply annulus mask by cut-out of image bad pixel mask to avoid counting bad pixels
- Mask other stars in annulus (crowded-field photometry)
- Ensure no other stars in circle!
- NOTE: Crucial to count pixels when finding the area, do not use πr^2

14.5 Or Else!: : — :, 5 min

- Saturn occultation using aperture photometry vs. raw star counts:
- `gv isofit_new.ps &`
- `gv isofit_oldb.ps &`
- `gv run_invert_jh_horizons_rgf_imlc.ps &`
- `gv run_invert_jh_horizons_rgf_imlc.V2.ps &`

14.6 Advanced 1: Aperture Correction: : — :, 5 min

- If you have *good* centering, a consistent PSF, and a good PSF model (e.g., spacecraft data)...
- You can take a smaller aperture, losing some star flux
- This reduces N_{sky} , and thus read noise
- Do photometry on the model PSF with the same aperture
- This finds the fraction of captured flux
- Divide measured photometry by fraction of captured flux

- This is tricky; do some tests to see if it improves things
- E.g., move aperture around on PSF with expected centering accuracy to see if variation exceeds benefit from small aperture

14.7 Advanced 2: Interpolated Aperture Photometry: : — :, 5 min

- For underresolved images, pixelation creates noise comparable to read noise or worse
- Use either nearest-neighbor or bilinear interpolation (they conserve flux)
- Interpolate each pixel to a 5×5 grid (test to see if more improves things)
- Do photometry on that
- Image size will jump by $25 \times$, so cut out a small area, first!

Chapter 15

Bad Pixels

15.1 Check In: : — :, 10 min

- Questions before we start?
- How much time spent in each HW?
- How is HW7?
- Debugging and “theoretical programming”

15.2 HW8 Peer Evaluation: : — :, 60 min

15.3 Bad Pixels and Cosmic Rays: : — :, 10 min

- Handle bad data (hot pixels, dead pixels, cosmic rays) by making and multiplying masks (MMM method)
- A mask is a Boolean array with same size as image
- In languages without a Boolean type, use smallest integer (e.g., byte)
- True=1=good, False=0=bad
- If possible, ignore masked data
- Else, “fix” bad pixels by interpolation (and tell people about it!)
- Make one mask for each input frame (flat, sky, object, dark)
- Median combinations count as 1
- Ignore median combinations if same effects in different frame
- AND or multiply masks to get final mask

- Of course, try to put your object on a part of the array away from bad pixels!
- Isolated pixels can use bilinear interpolation (draw)
- Regions of bad pixels: fit model to surrounding good data

15.4 Making Masks: : — :, 15 min

- `14photcal/badpixfilt.py`...
- Ideal: calculate data-minus-truth, flag anything not 0
- Instead of “truth”, use smoothed version of data (what eye does)
- Must account for noise and real data
- Ways to make a mask:
 - By hand (e.g., for array defects)
 - Flag pixels above/below levels that are outside good data (e.g., in flats)
 - Calculate median-filtered image:
 - * Each pixel replaced by the median in a box around it
 - * Size of box should be odd (centered on pixel)
 - * May need to embed data in larger array or set flags in Python routine
 - * `help(scipy.ndimage.filters.median_filter)` in Python
 - * This method is good for smoothly-varying sources (e.g., galaxies), may not work for spiky things like stars.
 - Calculate boxcar-filtered image:
 - * Each pixel replaced by the average in a box around it
 - * Size of box should be odd (centered on pixel)
 - * `help(scipy.ndimage.filters.uniform_filter)` in Python
 - * Sometimes better for stars, depends on seeing.
 - Calculate Gaussian-smoothed image:
 - * Convolve image with Gaussian to smooth each pixel into neighbors
 - * Width of Gaussian should not be more than a few pixels
 - * `help(scipy.ndimage.filters.gaussian_filter)` in Python
 - * Sometimes better for stars, depends on seeing.
 - Subtract this image from the real one, set levels, and flag.
 - Set levels by standard deviations from median
 - Set levels by inspection (usually better, done once per itime-filter combo)
- If making mask from something flat, use larger box and tighter limits

- Else, use small box and loose limits
- For mask from flat field, anything $0.5 < z < 1.5$ probably good
- Look at your masks, blink with data!
- If automating analysis (many images), test masks on many images!

15.5 Open Lab, Work on Project: : — :, 60 min

- Go around room and check in with each student
- Note where each student is (what step number) and tell me later
- Ask if anyone has a creative way to do any step, have them discuss
- Ask if anyone is stumped, have them discuss
- Ask if any undergrad has seen the transit, show
- Ask if any grad has seen the raw radial-velocity curve, show
- If there are questions, lead a discussion
- I will try to be on skype for questions for some of this time

15.6 Open Lab, Work on Project With Instructor Help: : — :, 45 min

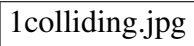
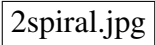
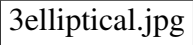


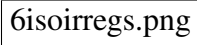
Chapter 16

Extended-Source Photometry and Atmospheric Calibration

16.1 Check In: : — :, 10 min

- Questions before we start?
- How was HW7?
- Who has read what? Quiz next Thursday unless all have read everything by Tuesday
- Reminder to take good notes in class for homework, watch videos of lectures

16.2 Extended-Source Photometry: : — :, 10 min

- Galaxies, nebulae, planets are not point-like 
- Galaxies: fit laws to radial dropoff in brightness 
- Find 50% enclosed light radius 
- Use standard stars to estimate flux per DN
- Convert images to magnitudes per sq. arcsec (basically take log & scale)
- Plot isophotes 
- Irregulars 
- Can find axis directions, etc. 
- Maps of planetary surfaces
- Atmospheric feature and wave tracking

- Atmospheric multi-wavelength cloud retrieval models
- All applications have specific methods to apply to calibrated data.

16.3 Photometric Calibration: : — :, 5 min

- We've corrected an image for both array effects and sky brightness
- We've measured the total DN for a star (and only from the star)
- These units are proportional to flux, but they're meaningless
- The same star will give a different number of DN later the same night! (Why?)
- Need to convert DN to standard flux units: i.e., calibrate

16.4 The Absolute Calibration Bootstrap: : — :, 10 min

- Flux (F) units:
 - $\text{erg} / (\text{cm}^2 \text{ sec})$
 - W/m^2
 - ADU/sec (relative unit)
- Note all are *rates*: always divide by exposure time
- Intensity (I) expressed per unit wavelength or frequency:
 - $\text{erg} / (\text{cm}^2 \text{ sec } \mu\text{m})$
 - $\text{erg} / (\text{cm}^2 \text{ sec Hz})$
- Do not cancel sec and Hz!
- Specific intensity is intensity per square arcsec
- Intensity integrated over all wavelength is the *bolometric* intensity
- Measured with a bolometer: black paint on an IR phototube or array to convert all energy into heat
- Absolute calibration is hard:
 - Sky transmission varies
 - Detector varies, detectors vary with respect to each other
 - It's hard to put a lab flux standard into a telescope beam
- Instead, do relative comparison to other stars

- Calibrate some stars well and bootstrap from them if really needed
- Vega (α Lyra) is *the* standard star
- Best calibration: $\sim 10\%$
- All other stars calibrated relative to Vega

16.5 The Magnitude System: : — :, 10 min

- Usually, we care about multiplicative differences, not additive
- How many times brighter is an F star compared to an M star, etc.
- We want a system that makes multiplicative calculations and relative flux comparisons easy
- It's called the magnitude system:

$$\Delta m = m_2 - m_1 = -2.5 \log_{10}(F_1/F_2) \quad (16.1)$$

- $\Delta m = 2.5$: factor of 10
- $\Delta m = 1$: factor of 2.5 (almost!)
- Factor of 10 = 2.5 magnitudes
- Factor of 100 = 5 magnitudes
- Dim stars have *higher* magnitudes!
- Vega has magnitude 0
- Sirius, Venus have negative magnitudes
- Young people with good eyes can see 6th magnitude in dark skies
- Sun has magnitude -26

16.6 Standard Star Calibration: : — :, 5 min

- Measure DN from target (I_{targ})
- Measure DN from standard (I_{std})
- Must do both measurements *under the same circumstances* (airmass, conditions, instrument, filter, etc.)!
- Standard star has known magnitude (m_{std}) relative to Vega
- Magnitude of target is:

$$m_{\text{targ}} = -2.5 \log_{10}(I_{\text{std}}/I_{\text{targ}}) + m_{\text{std}} \quad (16.2)$$

16.7 Atmospheric Correction: : — :, 10 min

- Sadly, it's rare to have a standard star in every frame, or even in any
- Transmission of the atmosphere depends on airmass
- `draw diagram of observer, atmosphere, star at different altitudes`
- Must somehow correct the atmosphere
- The mass of the air being traversed is the *airmass*, X
- $X \simeq \sec(z)$, where z is zenith angle
- `14photcal/airmass.py`
- Absorption really starts to pick up at 45°
- Go home at 70° !

- Atmospheric correction:

$$m_z = m_0 + aX \quad (16.3)$$

- a is the *atmospheric extinction coefficient*
- Measure a by plotting m_z vs. X
- What are the units? Magnitudes per airmass.
- Typical extinction values: less, except
 - UV: 1 mag/airmass or more
 - Optical: 0.2–0.3
 - IR: 0.1 or less
 - Any absorption band: high values!
- See Howell Table 5.1
- m_0 is an extrapolation: night must be very good to get
- Called a photometric zero-point
- If measurements are within $\sim 1\%$ of fitted line, night is “photometric”, zero-points are meaningful, and all celebrate!
- Measure both target and standard star several times in night, same filters, preferably close in time
- Use all data to fit a single a and Δm

16.8 Common Calibration Mistakes: : — :, 5 min

- Different filters
- Different photometry apertures
- Different instrument setting (gains, etc.)
- Differently-acquired flat field or dark data
- Different night (a varies)
- Forgot to divide by exposure time

Chapter 17

The Project

17.1 The Project: : — :, 15 min

- Final assignment is a project paper due in finals week
- Worth almost half of the grade
- Each project has you reduce and analyze some raw data, and write a paper
- Do the analysis exactly as a homework:
 - Main file with commands and comments on analysis
 - Generic analysis routines written for general use
 - Create figures and tables as needed
 - Include any tests
 - We will be running them from the shell with a `./project_sol.py` command
 - Make sure they work!
 - Project description says what to do, step by step
 - You may run into memory problems. You must find workarounds.
 - For example, there may be too much data for a median filter.
 - But, you can divide the data cube into quarters, do each, and sew the results together.
- Graduate project weighs the dark matter in a galaxy
- Undergraduate project is exoplanetary photometry
- Project analyses are harder than homework
- You must do all the work on the project yourself
- You may discuss techniques but not quantitative results
- You may not share code, look at each others' screens, etc.

- Paper should look like a research paper (e.g., *ApJ*):
 - Title (usually generic, serious, not flashy)
 - Author, affiliation
 - Abstract
 - Introduction: look on ADS and/or Simbad for at least 1 fact
 - Data Description (table good)
 - Data Correction
 - Measurement (name after actual measurement)
 - Results
 - Discussion
 - Conclusions
 - References (at least 1 on topic, 1 for methods)
- Can use Latex or own SW. There are latex templates online.
- No longer than 5 journal pages, please!
- Shorter is better but everything must be there. No gaps!
- This is a big deal. It will take 40 hours. START NOW!

17.2 Project Grading: : — :, 10 min

- Final course grade project components:
 - 13% Results: does the code produce all the right answers?
 - 5% Execution: how well is it coded?
 - 7% Paper: all aspects (completeness, writing quality, appearance, good use of plots and tables, references, error analysis, etc.)
- Execution: Project does not specify every routine like HW did
- You must use your judgement to design the analysis
- Follow rules of good coding from handout at start of class, HW comments, examples and commentary in solutions
- Generate all figures, tables, and numbers in your code
- Do error analysis in your code and explicitly in your paper as formulas
- Follow the `module_template_with_docstring.py` doc template to document each function

- Write generic functions for repeated and standard tasks
- Write separate, specific functions for this analysis
- Drive everything from a single, executable, top-level function
- Assign values to variables, then use those variables in function calls
- Indent and space the code internally to align similar things (why?)
- Use mnemonic variable and function names
- Avoid loops where possible
- Comment your code
- Test your code in small pieces!
- Hint: For large datasets, it may be faster to save after big operations
- Hint: For large datasets, it may be faster to test on subsets of data

Chapter 18

Interpolation and Correlation

18.1 Check In: : — :, 10 min

- Questions?
- How are projects going?
- Who will try to write the paper in LaTeX?

18.2 Interpolation: : — :, 15 min

- Interpolation is estimating a number based on other, related numbers
- Uses
 - Finding value of a function away from other data
 - Putting two (or more) datasets on a common grid
 - Reducing/increasing number of data points, resolution, etc.
- Caveats
 - It's a LIE! *Never* forget that.
 - Must always state in papers when using interpolated data
 - Statistical properties differ from uninterpolated data
 - Derivative properties differ from uninterpolated data
 - Extrapolation can lead to suffering! Try not to.
 - Interpolating interpolated data leads to junk. Don't.
- Try to avoid guessing or wholesale fabrication of data!
- Fit a function to data
 - May be piecewise: only to local data near interpolation point

- Fit may be overconstrained (use more points than needed), or not
- Fit may specify derivatives, or not:
- A jet flies at 600 km/h eastward. It passes Toledo at 5:29 pm. Where is it at 7:30?
- Evaluate function at point
- Linear: compute line between 2 points, evaluate at third
- Piecewise polynomial: fit cubic to 4 nearest points or parabola to 3 nearest
- Bilinear: 2D version of linear. First interpolate in x , then y .
- Spline: Piecewise polynomial with continuous derivatives
- Sinc: Theoretically perfect for band-limited signal

18.3 Cross Correlation: : — :, 45 min

- Cross correlation seeks the shift, s , between two (similar) datasets:

$$C(s) = \int_{-\infty}^{\infty} f(x+s)g(x)dx \quad (18.1)$$

- Evaluate for reasonable range of shifts, max is best alignment
- For spectrum shifts (e.g., Doppler), $x = \lambda$ and $s = \Delta\lambda$
- Discrete math, must generally:
 - Generally, want to use $f - \bar{f}$ and $g - \bar{g}$
 - Generally need to pad data (see demo)

- Note: This is different from the correlation coefficient!
- Correlation coefficient is dimensionless, ranges [-1,1]

$$\rho(x, y) = \frac{\sum_{i=1}^N (x - \mu_x)(y - \mu_y)}{N\sigma_x\sigma_y} \quad (18.2)$$

- It tests for trends (correlation) in sets of (x,y) datapoints
- Cross correlation measures shifts between two datasets in the same independent variable
- Assess the UNNORMALIZED correlation to do this
- Demo: `21crosscor`, start with `Correlation_examples.png`

18.4 Coding Notes: : — :, 20 min

- Killing a process:
 - Nothing in this class takes more than a few minutes to run, most runs in 1 second or less.
 - If something won't quit, kill it.
 - If you can get a prompt, type `jobs` and then `kill %num`, where *num* is a job number.
 - In another window, `ps aux | grep username`, then `kill processnum`, where *processnum* is a process number from the second column of `ps aux`.
 - `top` and `htop` give the most CPU-using processes.
- Memory use:
 - Record floats, not doubles, usually (as FITS)
 - If short on memory, use floats for image data (why?)
 - Please do not hand in files not requested
 - Work on stuff in place
 - Delete variables
 - Use `top`, `htop`, and `ps` to monitor memory
- Careful passing an array slice, different IDL vs. Python
 - IDL: Gets turned into anonymous object
 - Gets deleted at end of call
 - Therefore, will not be modified in caller
 - Python: A *view* into the array is passed
 - Modifies array *slice* in caller
 - If routine will modify and that's not wanted, pass a copy of the slice (or other variable)
 - Demo: `15spec/slicepass.py`
- Debugging: Test your output!
 - Make trivial test cases, run routine, check answer
 - Check that what you got back is what you expect on real data
 - At least check that the result is sane (not all same value, etc.)
 - Try taking each thing you calculate (including numerators and denominators separately, etc.) and doing it separately, then look at what you get and see if you get what you want. Is the result the right shape? Ballpark the right numbers?
 - Try some simple, small test cases. Take your object array and set the entire first image to 1, the whole second image to 2, and so on. You know what the medians are already. What do you get at each step? It should be easy to verify correctness and find problems.

- Use course staff
- Quote on debugging:

I went on to test the program in every way I could devise. I strained it to expose its weaknesses. I ran it for high-mass stars and low-mass stars, for stars born exceedingly hot and those born relatively cold. I ran it assuming the superfluid currents beneath the crust to be absent – not because I wanted to know the answer, but because I had developed an intuitive feel for the answer in this particular case. Finally I got a run in which the computer showed the pulsar’s temperature to be less than absolute zero. I had found an error. I chased down the error and fixed it. Now I had improved the program to the point where it would not run at all.

— George Greenstein, “Frozen Star: Of Pulsars, Black Holes and the Fate of Stars”

Chapter 19

Introduction to Spectroscopy

19.1 Check In: : — :, 5 min

- Questions before we start?

19.2 What is Spectroscopy?: : — :, 10 min

- *Photometry* answers: How bright?
- Specifically: How bright in a certain *wavelength bandpass*.
- Usually done with filters on an array
- *Spectral resolving power* $\lambda/\Delta\lambda \sim 4-5$.
- For example, a 100-nm-wide filter at $\lambda=500$ nm.
- Much can be done very efficiently this way:
- Stellar temperatures, variable stars, etc.
- *Spectroscopy* answers: How bright as a function of wavelength?
- $\lambda/\Delta\lambda \sim 100-1,000,000$.
- Photometry measures one bandpass at a time.
- Spectroscopy measures many (10–10,000) small bandpasses simultaneously!

19.3 Spectroscopy Basics: : — :, 5 min

- Use *dispersion* to sort photons by wavelength
- This spreads a point source over a line

- Position on the line is related monotonically to wavelength
- Dispersers:
 - Prism
 - Diffraction grating
 - Grism (prism with grating etched onto one or both surfaces)

19.4 Overview of Spectroscopy: : — :, 25 min

- 15spec
- Add to part on prisms in first few slides:
- Snell's law:

$$n_1 \sin \theta_1 = n_2 \sin \theta_2 \quad (19.1)$$

- n_x are indices of refraction
- θ_x are normal angles
- θ_1 and n_1 are (nearly) constant
- n_2 varies strongly with λ
- So, θ_2 changes with λ
- Prisms have low dispersion, $R = \lambda/\Delta\lambda \ll 1000$

19.5 Quiz 3: : — :, 20+10 min

19.6 Go Over HW9: : — :, 45 min

19.7 Spectral Observing Modes: : — :, 5 min

- Long-slit spectroscopy
- Echelle (cross-dispersed) spectroscopy
- Integral-field spectroscopy
- Fiber-feed spectroscopy 15spec/3-fiberdiag.jpg

19.8 Spectroscopy Examples: : — :, 5 min

- [15spec/4-starspec.png](#)
- Continuum shape gives type of emission
- Blackbody temperature
- Lines give atomic/molecular composition
- Line widths give temperatures, pressures

Chapter 20

Spectroscopic Gratings and Extraction

20.1 Check In: : — :, 5 min

- Questions?

20.2 Spectrograph Optics: : — :, 25 min

- Collimated beam at grating, or beam and dispersion angles mixed
- Review of making a spectrum from an image
- Input image now smeared in λ
- R determined by PSF size: “true” spectrum convolved by PSF
- BUT, we can pass light through a *slit* to reduce PSF
- Spectrograph optics
 - Discard light, but improve R
 - Avoid nearby objects contaminating spectrum
 - Can use mirrored slit, send rest of image to slit-viewing camera
 - Camera can guide exposure
 - Camera can monitor slit placement

20.3 Slit Width and Resolving Power: : — :, 5 min

- The *resolving power* $R = \lambda/\Delta\lambda$
- $\Delta\lambda$ is set by the slit width: A wider slit integrates over more wavelength
- This increases $\Delta\lambda$ and reduces R

- Usually the slit is 3–10 pixels wide: want to overresolve pixel scale
- R is related to the *dispersion*, $d\beta/d\lambda$: High dispersion is high resolving power and less *grasp*
- Deflection angle β translates to distance in pixels in camera

20.4 Long-Slit Spectroscopy: : — :, 35 min

- `16specextract`
- Slit usually covers entire detector (why else?)
- Star is only part of the slit
- Can get sky background spectrum next to star
- Extended objects bigger, may or may not fill slit
- Scan slit: spatial-spectral hypercube
- Can also allow planet to rotate under slit

20.5 Gratings: : — :, 10 min

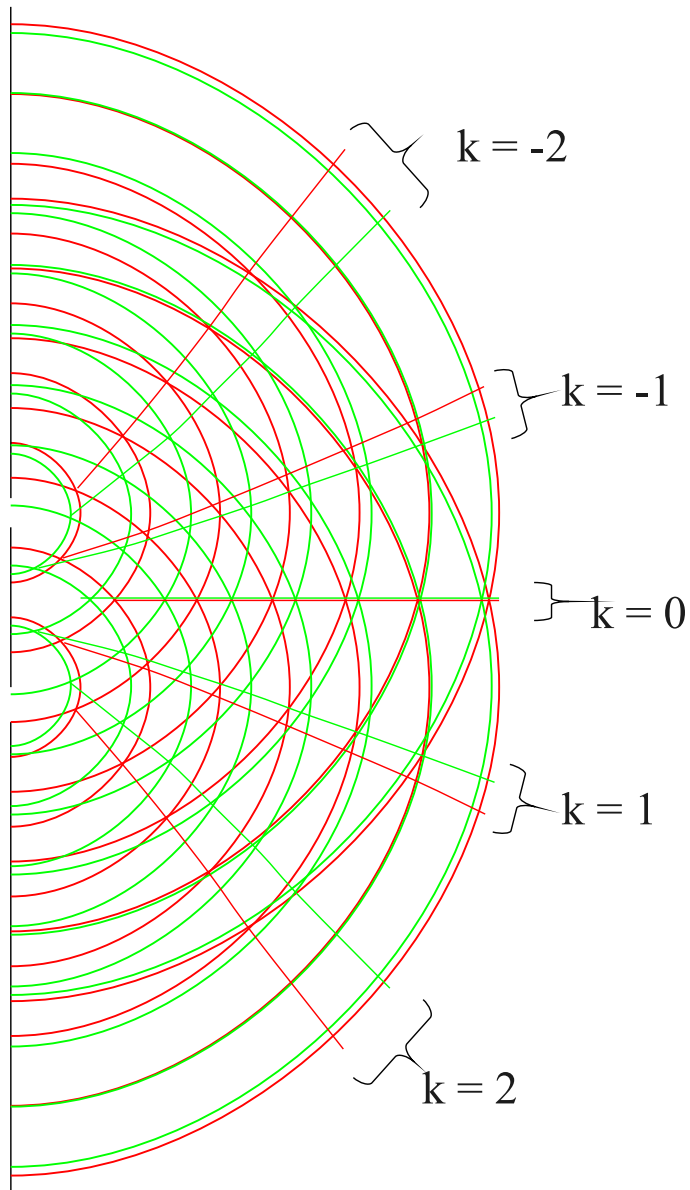
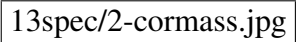


Diagram credit: Bob Macke.

- Diffraction Grating

- Piece of glass etched with fine lines
- Works interferometrically, not refractively
- At a given angle, certain wavelengths add, others cancel
- Different orders are different interferences
- Grating equation:

$$\sin \alpha + \sin \beta = nk\lambda \quad (20.1)$$

- n is blaze frequency, usually in lines/mm
 - k is order number
 - α is incidence angle from normal
 - β is diffraction angle from normal
 - Changing α (grating tilt) changes projected n
 - Also changes center λ and dispersion
- Grating equation: orders at different wavelength overlap
 - “Order sorting” done with filters
 - Or, can cross-disperse:
 - Low-dispersion element spreads light in one direction
 - High-dispersion element spreads light in other direction
 - 

Chapter 21

Spectroscopic Systematic Corrections

21.1 Check In: : — :, 5 min

- Questions?

21.2 Things to Correct: : — :, 5 min

- Array detector effects
- Sky emission (continuum and lines, especially OH⁻ lines)
- Sky absorption (mainly H₂O, some CH₄)
- Before, these two were just a part of the sky background and instrument throughput
- Now they vary with wavelength

21.3 How to Correct Them: : — :, 5 min

- Bias, dark current: dark frames
 - No light enters slit
 - Exposures must match target: wavelength-dependent effect!
- Gain variations: dome/screen flats (sky has lines)
- Sky emission (background):
 - Subtract off-target rows (or fit low-order polynomial across slit)
 - Extended object: subtract off-target exposure
- Sky absorption:
 - Divide by “spectral flat field” (featureless star, e.g., B or A star)

- Try to do low-, medium-, and high-airmass observations of same star, compute extinction coefficient per wavelength

21.4 Frames to Acquire: : — :, 5 min

- Target
- Dark
- Dome flat
- Sky (if necessary)
- Spectral flat
- What's left? Wavelength standard.

Chapter 22

Spectroscopic Wavelength Calibration

22.1 Wavelength Calibration Basics: : — :, 25 min

- Would like to know the wavelength, λ , falling on each pixel (x, y)
- That is, we want to find $f(x, y) = \lambda$
- For unresolved sources (stars), want $f(y) = \lambda$ for that observation
- Dispersion on array usually close to linear, but not perfect
- Offsets in starting λ per frame and along slit (“smile”)
- Take spectra of a source with known spectral lines
 - Could be lamp, sky, or lines in object spectrum
 - Lamp usually He, Ne, Ar, Xe (lines very precisely and accurately known)
- Identify line locations on the frame per row, or in spectrum draw, table of pix vs. λ
- Fit some function (usually polynomials) to those locations draw plot
- This maps out the array or spectrum
- For best fit, need many more lines than degree of polynomial
- Want some lines near each end of spectrum to avoid extrapolation
- Lines must be distinct for a good fit
- Lines must be narrow enough for project goals
- Is it always just a simple linear function?
- No: cross-dispersed spectra are 1 example 13spec/2-cormass.jpg

- Optics have non-linear dispersions

$$\frac{d\beta}{d\lambda} = \frac{d\frac{n_1}{n_2} \sin \alpha}{d\lambda \cos \beta} \quad (\text{refraction}) \quad (22.1)$$

$$\frac{d\beta}{d\lambda} = nk / \cos \beta \quad (\text{diffraction}) \quad (22.2)$$

- NOTE: n is index of refraction in Eq. 22.1 and blaze frequency in Eq. 22.2
- Dispersion is usually close to linear per row
- But, zero of each row may differ (“smile”)
- Stars shift in slit frame-to-frame!
- Shift wavelength scale by fractional pixels to correct

22.2 Example Using OH Lines: : — :, 30 min

- `18wavlcal`

22.3 HW10 Peer Evaluation: : — :, 45 min

22.4 Calibration Sources: : — :, 10 min

- Common calibration sources:

Source	Wavelength	Resolution	Doppler	Slit Shifts	Extra Time
Object	any	depends on object	incl.	incl.	no
OH ⁻ sky lines	near IR	R>600 to resolve blends	no	no	no
Lamp: He, Ne, Ar, Xe	UV–near IR	all, watch blends	no	no	yes
Planetary nebulae (H, He)	visible–mid IR	low only (few lines)	extra	maybe	yes
Star lines	UV, near IR	depends on line	extra?	extra	yes
Telluric lines in starlight	near IR	depends on line	no	extra	yes

22.5 Calibration Observations: : — :, 5 min

- How often?
- Often enough to make sure the “solution” (fit) has not changed *significantly*
- Depends on:
 - Instrument mechanics (flexure)
 - Instrument temperature changes
 - Resolution
 - How accurate a calibration the program requires (i.e., what’s “significant”)

- `19specflex`

Chapter 23

Spectroscopy Applications

23.1 GTA Evaluation: 10:05 — 10:15, 10 min

NOTE: ALL questions are about the GTA, even where it says “instructor”. You will evaluate the professor at the end of the semester.

Week 9.

23.2 Check In: : — :, 10 min

- Questions?

23.3 Gravitational Redshift: : — :, 5 min

- Photons leaving a massive body increase their potential energy
- This decreases their kinetic energy
- Since $E = h\nu$, if E drops, so does ν
- So, they shift to longer wavelength

23.4 Doppler Shift: : — :, 5 min

- If wave source moves toward observer, wave is perceived at higher frequency:

$$z \equiv \frac{\Delta\lambda}{\lambda} \approx \frac{v}{c} \quad (23.1)$$

- At high v :

$$\frac{v}{c} = \frac{(z+1)^2 - 1}{(z+1)^2 + 1} \quad (23.2)$$

- Wavelengths shift shorter for approaching source (blue shift)

- Wavelengths shift longer for receding source (red shift)
- If a 1- μm line appears at 7 μm , $z = 6$, $v = \frac{48}{50}c$
- Measure speed of Hubble flow for distant galaxies, reflex motions of stars hosting exoplanets, velocity of students from exams, etc.

23.5 Spectral Line Profiles: : — :, 5 min

- At high R , spectral lines have a shape, the *line profile*
- Several effects combine: some intrinsic and instrumental
- Instrumental: mainly the finite slit width
- Intrinsic: nature has no δ functions!
- Line broadening comes from: (Don't go into detail, that's next.)
 - Temperature (Doppler) broadening
 - Lifetime/collisional broadening
 - Zeeman (magnetic field) and electric field broadening
- Final profile is convolution of all independent effects

23.6 Electrical and Zeeman Broadening: : — :, 5 min

- Effects that shift or split quantum levels also shift line wavelengths
- EX1: High density/pressure distorts orbital shape and energy levels
- EX2: Zeeman effect: high magnetic fields (e.g., near surfaces of neutron stars or in sunspots) split and shift quantum states
- Can measure amount of split, and thereby directly measure field strength in a remote object

23.7 Temperature Broadening: : — :, 5 min

- Molecules in a gas move, $E \propto kT$
- $mv^2/2 \propto kT \rightarrow v \propto \sqrt{2kT/m}$
- Molecules move in all directions, with a distribution of speeds around v
- Each molecule's lines Doppler-shifted according to its thermal motion
- Lines broaden according to T

- Thermal line-broadening profile is a Gaussian:

$$\alpha_D = \frac{\nu_0}{c} \sqrt{\frac{2kT}{m}} \quad (23.3)$$

$$p_D(\nu) = \frac{1}{\sqrt{\alpha_D^2 \pi}} \exp - \left(\frac{\nu - \nu_0}{\alpha_D} \right)^2 \quad (23.4)$$

23.8 Lifetime/Collisional Broadening: : — :, 10 min

- Molecules excited to high energy state spontaneously decay to low state at some rate
- OR, de-excited in a collision (emitting a photon)
- 2 effects combine, average lifetime of state is Δt
- Line gets broad when Δt is small compared to $1/\nu$
- Heisenberg Uncertainty Principle:
- $\Delta p \Delta x \geq \frac{\hbar}{2}$
- $\Delta E \Delta t \geq \frac{\hbar}{2}$
- But, $E = h\nu$. We observe ν so we must know E !
- But we can't know E precisely, so there must be a spread of ν .
- A solution to the Schrödinger equation gives the Lorentzian profile, width is α_L

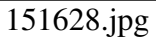
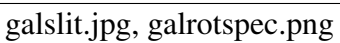
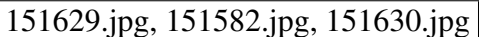
$$f(\nu - \nu_0) = \frac{\alpha_L}{\pi [(\nu - \nu_0)^2 + \alpha_L^2]} \quad (23.5)$$

$$\alpha_L = \frac{1}{4\pi \Delta t} \quad (23.6)$$

$$f(\lambda - \lambda_0) = \frac{\alpha_L}{\pi \left[c^2 \left(\frac{1}{\lambda} - \frac{1}{\lambda_0} \right)^2 + \alpha_L^2 \right]} \quad (23.7)$$

- α_L related to Einstein absorption coefficient $A_{lu} = 1/\Delta t$

23.9 Dark Matter: : — :, 15 min

-  We measure rotation by Doppler shift
-  Place slit on galaxy axis, get spectrum vs. r
-  Rotation curve

- Enclosed mass formula:

$$F_c = \frac{mv^2}{r} = \frac{GMm}{r^2} \quad (23.8)$$

$$M = \frac{v^2 r}{G} \quad (23.9)$$

- $v=220$ km/sec, $r=28,000$ ly
- Milky way: $M_r = 2 \times 10^{41}$ kg = $10^{11} M_\odot$
- $10^{11} M_\odot$ w/in Sun's orbit
- But, we measure constant v with increasing r
- This means increasing M , even though decreasing luminosity: dark matter
- Other applications of Doppler shift include finding extrasolar planets, measuring the velocities of stellar winds, and measuring planetary zonal winds
- Use cross correlation to find shift

Chapter 24

The Project Paper

24.1 Paper Content : — :, 10 min

- Project/course focus is on data analysis, so put most emphasis there.
- Describe basic things in one sentence.
- Do not explain standard things like flat-fielding.
- Do explain any judgement calls, unusual procedures, expediences.
- Do not describe each line of your code (but DO comment your code well)
- Spectrum papers will be much longer than transit papers: more steps
- Use figures to present milestone analysis steps
- Do not rehash instructions; summarize except where explanation needed

24.2 Paper Format : — :, 10 min

- Would like to have paper look somewhat like a published paper
- ApJ author information: <http://www.iop.org/EJ/apj>, click on “Submit an article”, see links on left
- Don’t waste time on formatting details (fonts, exact sizes, etc.)
- Get and read a few short ApJ papers (use ADS or go to ApJ site)
- Standard sections of a research paper:
 - Title
 - Author(s) and affiliation(s)
 - Abstract

- Introduction
 - Observations
 - Data analysis
 - Discussion
 - Conclusions
 - References
- Obs-anal-discussion are often broken down differently, depending on the paper, but the topics appear in this order
 - Best to keep hard facts and the interpretation separate
 - Use subsections as needed
 - Reference format
 - Table format
 - Figure and caption format

Chapter 25

Probability and Error Analysis Review

25.1 Selection of Review Topics: : — :, 10 min

- Next topics are optimal extraction and Markov-Chain Monte Carlo error estimation
- Will skip Fourier transforms
- Need some review first. S/N questions being missed by all.
- Topics for review:
 1. Types of error
 2. Probability terms (draw, sample, population, estimate, etc.)
 3. Probability properties (PDF, cumulative, integrate over interval, etc.)
 4. Probability formulae (expected value, discrete vs. continuous)
 5. Probability distributions (uniform, Gaussian, Poisson)
 6. Confidence interval
 7. Error analysis and probability
 8. Error propagation equation
 9. Error propagation in +, -, \times , \div
 10. S/N
 11. **Noise sources in astronomical data**
 12. Astronomical S/N estimates
 13. Model fitting basics
 14. Least squares
 15. Fitting analytic models (least squares line fit)
 16. Fitting with a minimizer

25.2 Review: : — :, 90 min

Chapter 26

Detector Physics Review

Take the sections from CCD and IR array, but plan 60% of the time. Have them describe how a CCD and an IR array work, then show the lecture. This should take half a class.

Chapter 27

General Model Fitting

27.1 Check In: : — :, 10 min

- Questions?
- How are projects going?
- Who will try to write the paper in LaTeX?

27.2 Using Models to Interpret Data: : — :, 5 min

- Quick review to get us introduced to Markov Chains
- We take data with a goal in mind, e.g.:
 - Test a theory
 - Do measurements of several objects do what theory says?
 - Assign a numerical value and an error to a parameter for a given object
 - E.g., find the extinction for this night
 - Or the center wavelength of this line
- Sometimes we read the answer right from the data
- For example, Doppler shift of a quasar $z = \Delta\lambda/\lambda$
- More likely, however, we fit a model to the data (even for Doppler)
- Fitting is the most precise way to interpret digital data
- WE DO NOT FIT THE DATA TO A MODEL!
- So far we've done several types of model fits
 - Linear least squares (derived formulae)

- General linear fitting (theory vs. data w/ multiply and shift)
- Function minimization fitting
- Gaussian and polynomial fitting using function minimizer
- Problem: if parameters are correlated, error estimates are wrong
- For complex models, some parameters are generally correlated

27.3 Least Squares: : — :, 10 min

- We start with data and a model we wish to fit
 - Use model to predict data values
 - E.g., for each wavelength, calculate the amplitude to make a spectrum
 - Or for each time, calculate a photometric intensity
 - Then, do some evaluation of the quality of fit: just a single number!
 - Twiddle free parameters until the quality number is maximized

- What is a good quality number?
- Earlier we used the maximum likelihood method to justify least squares
- Want to minimize:

$$\chi^2 = \sum \left[\frac{y_i - f(x_i)}{\sigma_i} \right]^2 \quad \text{least squares} \quad (27.1)$$

- Recall that this assumes a Gaussian error distribution
- For some models, e.g., lines, can compute the free parameters from the input data
- For others, analytic solution may not be possible, or may be hard to derive: do it numerically

27.4 Project One-on-One Help: : — :, 55 min

Chapter 28

Optimal Extraction

28.1 Optimal Photometry: : — :, 30 min

- We could do photometry better:
 - Don't count as far out for dim stars, just count part of flux and adjust for known undercount (like census doesn't)
 - Or, do weighted sum, giving less weight to distant pixels
 - Fit a sloping background better
 - Don't use interpolated values from bad pixels

- Theory:

$$F_{*,\text{tot}} = \frac{F_{*,\text{pix}}}{p_G(x, y | \sigma_x, \sigma_y, x_0, y_0)} \quad (28.1)$$

- Compute this, and use optimal weighting for an average to many σ distance from x_0, y_0
- Optimal weighting:

$$\sigma_{*,\text{tot}}^2 = \frac{\sigma_{*,\text{pix}}^2}{p_G(x, y | \sigma_x, \sigma_y, x_0, y_0)^2} \quad (28.2)$$

- Must also add $\sigma_{\text{sky}}, \sigma_{\text{readnoise}}$, etc. in quadrature to get σ_{pix}
- Now, find mean with these error estimates
- If star noise dominates (bright star), this is like aperture photometry
- If sky noise dominates (dim star), this is Gaussian weighting
- NOTE: This method is used lots of places: spectrum extraction, etc.
- PSF isn't really Gaussian. Must have good, oversampled PSF image. Must center well.
- Same result from fitting a model, same probs.
- 24optphot

Chapter 29

Presenting Data: Plotting Well

29.1 Check In: : — :, 5 min

- Introduce yourself
- Questions before we start?

29.2 Video by Hans Rosling: : — :, 25 min

- One of the most popular TED talks is a data presentation by Hans Rosling.
- During the presentation, ask yourselves:
- What are the main points? Why does Rosling make a TED talk?
- Which point is he most concerned about?
- How does he use the non-data elements of his presentation? (What does he have besides his computer?)
- What plotting methods does he use to show more than 2 dimensions of data on 2D plots? Make a list as you watch.
- Note how simple and yet data-rich his plots are!
- 21datapres https://www.ted.com/talks/hans_rosling_shows_the_best_stats_you_ve_ever_seen

29.3 Discussion of Video by Hans Rosling: : — :, 20 min

- What are the main points? Why does Rosling make a TED talk?
- Which point is he most concerned about?
- How does he use the non-data elements of his presentation? (What does he have besides his computer?)

- Reviewing the video, what plotting methods does he use to show more than 2 dimensions of data on 2D plots?
- What methods does he use to show relationships in money, population, health? How does he show clustering, changes over time, changes with respect to members of a like group?
- What are some other methods you can think of to present data in 2D?
- Always keep an eye out for interesting ways to present data, catalog for your future use
- Just reading the matplotlib manual gives new ideas of how to present

29.4 Quiz 3: : — :, 20+10 min

Give them 20 minutes or until everyone is finished. You can give them longer if they need it. Collect papers. Then, have them come to the board to do each problem, and talk them through.

29.5 Rules for Plots: : — :, 25 min

- Know what you want your viewer to learn from your plot
- Know how your viewer will be seeing your plot (screen, print, projected, “live” on web)
- Know with what knowledge your viewer will be seeing your plot (after reading something? before?)
- Know how long your viewer will have to look at the plot
 - As long as desired (as in a paper)
 - Less than a minute (in a talk)
 - How much you put on a plot and the complexity of the idea is limited by the viewing time
- Let this information drive all your decisions
- Don’t do *anything* without a purpose for *this* audience
- KEEP IT SIMPLE, AVOID CLUTTER
- Have a title, axis titles, values on axes
- In a paper, the plot title is a sentence fragment that starts the caption. To save space, it is not located above the plot.
- “Size” is according to *angular size* as seen by viewer
 - Plot is “small” for slide show (screen is small for those in back), big for paper/computer screen (taking up 45° of visual field or more).

- Choose font sizes, styles, line thicknesses, point sizes accordingly
- Data are usually points; connect with lines only if that adds info
- Models are usually lines; if using points, make distinct from data
- Present multiple datasets only to compare them
- Keep keys far out of the way, or eliminate
- Do not use a grid (or make it very light, if critical)
- Keep data separated from axes on sides, top, *and* bottom, so reader can visually extrapolate
- Avoid distracting backgrounds, line styles, gratuitous visual effects, etc.
- Use shape, linestyle, or color to deliver information (only)
- If needed, highlight important information with thicker lines, brighter color, etc.
- Do not adjoin plots with unrelated axes
- Show it to someone like your audience without comment, see if they get your point
- Whatever they say, they likely represent others like them. Revise until they get it.
- Do this with multiple people.
- [goodplotbadplot.pdf](#)

29.6 Your Work: : — :, 15+5 min

If you can, give them a short presentation on your work, or some topic of interest to you. Take questions at the end.

Chapter 30

Presenting Data: Writing a Paper

30.1 Check In: : — :, 5 min

- Introduce yourself
- Questions before we start?

30.2 Learning to Write Research Papers: : — :, 5 min

- Today we'll talk about writing research papers
- They are different from lab reports, but a lot is similar
- It is a significant leap from creative or journalistic writing to research writing
- Respect that gap and focus on it, and you'll learn quickly
- The best way to prepare for writing papers is to read papers, focusing on the writing as well as the content
- Identify the writing style common to all papers
- See what you need to explain and what you can assume
- Look at the kinds of arguments raised, when to cite other work, what is in figures and tables, etc.
- Each journal has a style guideline, usually detailed, that you must follow. Read a few of these along with a paper from each journal.
- We'll start, today, by discussing the Deming *et al.* (2005) Nature paper, which reports the first detection of exoplanet photons

30.3 Discussion of Deming *et al.* (2005): : — :, 20 min

- Summary of paper (ask students)
- 3 rules of Communication:
 1. Know your subject.
 2. Know your audience.
 3. Know what you want your audience to know about your subject.
- What is the paper about?
- Who is the audience of the paper?
- What do the authors want to accomplish by publishing the article?
- What do the authors present, and in what order?
- What do they need to do to convince people it's right?
- How did they do that?
- Discuss the paper and the writing techniques used

30.4 Parts of a Research Paper: : — :, 15 min

- Each paper tells a narrative story, the argument that supports the paper's conclusions
- Writing the first draft early in the research can help to drive the research toward a good paper
- If you planned your research poorly, you may find yourself doing many unexpected calculations because your paper needs them!
- So, plan a story from the beginning
- A common set of sections for an observational paper is:
 - Abstract: stands alone on abstract services, summarizes the work and results, usually one long paragraph
 - Introduction: a “funnel” that interests a wide swath of readers' minds and prepares them to read the paper
 - Observations: lists and describes the data and their acquisition
 - Data analysis: describes how the data were processed to get numbers and uncertainties
 - Results (often named after the thing the data are about, such as “Orbit” or “Atmospheric Chemistry”): uses the numbers to constrain a theory
 - Discussion: talks about implications and limitations, compares to other work, etc.

- Conclusions: NOT just a summary, talks about what this work enables, future work, etc.
- References: standard-format (for journal) entries for all cited literature

30.5 The Abstract: : — :, 5 min

- The abstract is all most people will read!
- It appears on preprint servers and in literature searches
- Summarize the goal, the approach, the result
- Include the key numerical results (with uncertainties) that people might look for in the papers
- Be terse, do not expand on anything
- State any key claims
- Describe anything major but not yet mentioned that the paper covers
- NOTE: *Nature* letters (only) do not have a real abstract. They have a “bold paragraph” that mostly serves this function. It has a specific format and might not state numerical results.

30.6 The Introduction: : — :, 10 min

- Collect a variety of minds, not all of them in the field, interest and focus them on the work
- Justify the need for this work
- What question are you addressing?
- Who will use the result?
- Introduce background literature
- Lay the stage by discussing prior studies and related work
- State your approach to the work
- Give a “roadmap” to the rest of the paper

30.7 The Main Text: : — :, 10 min

- The style is terse. Space is at a premium. So is the reader's time.
- Do not explain things that are generally known
- Keep the discussion at a high level
- E.g.: Do not explain what a flat field is. Rather, say your flat fields came from the difference between sky images at two different light levels, for example.
- Keep the sections clear, and clearly organized
- Examine each sentence and ask, can I take this out? Each sentence should have a clear function.
- Use short expressions: “because”, not “due to the fact that”
- But, make it no shorter than it needs to be. You will have to develop judgement in this area by reading other papers.
- Cite any argument or fact that comes from somewhere else
- Cite all relevant research, or cite a review paper “and references therein”

30.8 The Conclusions: : — :, 10 min

- **This is not a rehash of the abstract!**
- Summarize the claims and in briefest terms what you did to prove them
- Then, talk about what this now enables
- Who cares about this result?
- Talk about the future, what the next steps are, what might be needed to get a better result or the next result
- Finish with a strong sentence

30.9 Paper Writing Style: : — :, 10 min

- Paper style is unique, different from textbook or creative writing
- You are writing for history. You are writing *to* an editor with a budget of pages.
- High-level. Serious. Unemotional. Minimalist. Concise!
- Active voice, but avoid repeating “I”/“We”

- Think of what did the action and use that as the subject: “A quadratic fit the shifts...” rather than “A quadratic was fit to the shifts”.
- *Definitely* avoid a passive sentence ending in “by the author”!
- Avoid references to your mental state or effort unless it affected the result (“this was hard”, “after 27 hours I was finally able to”, etc.). Such references do not appear in professional work.
- Use simple sentence structures and grammatical constructs. Many of your readers will not be native English speakers. Give them a break!
- Engineer each sentence to a purpose. Always ask:
 - Can this be made shorter without losing information?
 - Can this be made simpler?
 - Exactly what information do I want to convey with this sentence?
 - What information does this sentence need to be understood?
 - What sentences need this sentence to be understood?
 - What is the role of this sentence in the paper?

30.10 Discuss Papers Class Read: : — :, 20 min

- Each student read another paper. Have each student talk for 1 minute about the paper and one writing-related item that struck them. You can use the document camera or put the PDF on screen via a flash drive or a web browser on your laptop or on the podium computer.

Chapter 31

Bayesian Statistics and Markov Chains

31.1 Two Approaches to Statistics: : — :, 5 min

- Two fundamental approaches to the statistics of fitting models to data:
- **frequentist**: How frequently would data better than ours (lower χ^2) come from this theory?
- **Bayesian**: What is the probability of the theory given the data?
- The second approach is obviously the question we want to answer
- But, must calculate χ^2 for tens of thousands of theories similar to a candidate theory to see how likely the candidate theory is!
- This was not tractable before computers, so frequentist stats were invented:
- Given χ^2 and number of degrees of freedom, can calculate the probability of getting better data in the next experiment.
- This has been the main approach to statistics for the past 2 centuries.
- But, computers now allow us to use the Bayesian approach.

31.2 Conditional Probability: : — :, 10 min

- What does “Bayesian” mean?
- Our question said, “What is the probability of the theory **given** the data?”
- If it has the word “given”, it’s conditional probability.
- Bayes’ theorem of conditional probability:

$$P(a, b) = P(b)P(a|b) \quad (31.1)$$

$$P(a|b) = \frac{P(a, b)}{P(b)} \quad (31.2)$$

- a and b are events or outcomes
- $P(a, b)$ is the probability of a and b both happening
- $P(a|b)$ is the probability of a happening after b has already happened (probability of a given b)
- Try it with coin tosses (independent a and b) for an unfair coin with $P(a) = 1/3$ and $P(b) = 2/3$
- Since a and b are independent, $P(a) = 1/3$ at any time
- $P(a, b) = P(a)P(b) = 1/3 \cdot 2/3 = 2/9$
- Check: $P(a|b) = \frac{P(a,b)}{P(b)} = \frac{2/9}{2/3} = \frac{2}{9} \cdot \frac{3}{2} = 1/3$
- Conditional probability is simple, but very deep.
- It can be applied to some seemingly non-math situations.
- Laplace, a mathematician and judge, applied it in jurisprudence and governance:
- What is the probability of guilt given the evidence?
- How should we spend our budget given the challenges?

31.3 Bayesian Statistics: : — :, 10 min

- If we calculate both $P(a|b)$ and $P(b|a)$, we can eliminate $P(a, b)$:

$$P(a, b) = P(b)P(a|b) \quad (31.3)$$

$$P(a, b) = P(a)P(b|a) \quad (31.4)$$

$$P(b)P(a|b) = P(a)P(b|a) \quad (31.5)$$

$$P(a|b) = \frac{P(a)P(b|a)}{P(b)} \quad (31.6)$$

- If a is a model (or a vector of parameters to a model),
- and if b is the data, then:
- $P(a|b)$ (probability of model given data) is the “posterior distribution”, the distribution as we know it after the calculation.
- $P(a|b)$ has as many dimensions as there are free parameters in a .
- We can generate histograms of parameter probabilities and their uncertainties from the posterior.

- $P(b|a)$ is the probability of the data given the model. This is proportional to $e^{-\chi^2/2}$ for Gaussian, independent errors. We saw this quantity in least squares.
- $P(a)$ is the “prior knowledge” of the model’s parameters. A previous observation may constrain our fit.
- $P(b)$ is the prior knowledge of the data. Often this is unknown and is eliminated, turning the formula into a proportionality.

31.4 Mapping Phase Space: : — :, 5 min

- Consider a linear model (i.e., 2 parameters).
- $P(a|b)$ is planar and the good solutions are likely in one region of the plane.
- Make a map of χ^2 vs. the 2 parameters by gridding and calculating.
- The shape of the χ^2 minimum gives the parameter fits and their uncertainties.
- For more parameters, however, calculating quickly becomes difficult.
- Enter Markov-Chain Monte Carlo.

31.5 Stevenson MCMC Demo: : — :, 25 min

Chapter 32

Postmortem

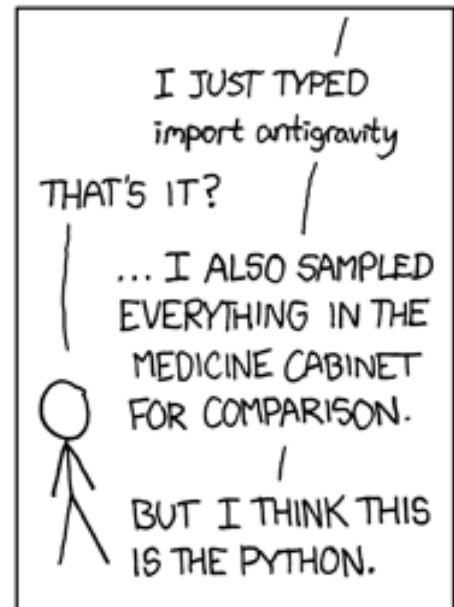
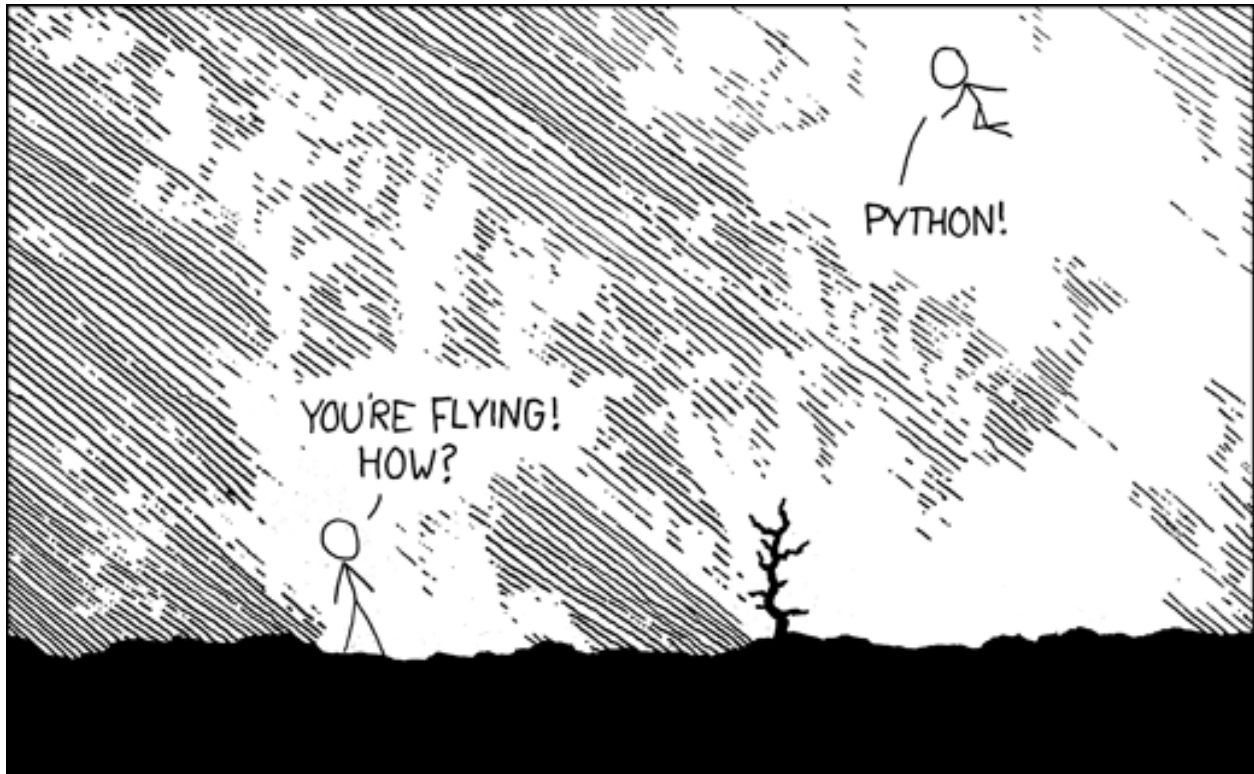
32.1 Check In: : — :, 5 min

- Questions before we start?
- Monday 2 Dec, PSB 441, 10:00 – 6:00 (mostly), project help
- Tuesday 3 Dec, PSB 441, 10:00 – 12:00, project help
- Wednesday 4 Dec 1:00 pm project due (change of time from syllabus)
- Final exam period: 10 Dec, MSB 306, 10:00 – 12:50, project discussion, THIS IS AN OFFICIAL CLASS!
- PLEASE fill out the course evaluations online at my.ucf.edu!

32.2 Quiz 6: : — :, 25 min

32.3 Class Postmortem: : — :, 20 min

- Classes are starting point for real work
- You should, at this point, feel ready to learn more about programming, data analysis, and astronomical measurement on your own.
- You should be confident asking any astronomy researcher for a job doing entry-level work.
- Grads should be ready to work on their own projects with advice of a more experienced researcher.
- Do you feel that way? How did the course meet or fail to meet your expectations?
- Should I make progress on the project a homework assignment?
- My eval form



***** START OF STUFF TO CUT *****

Chapter 33

Random Sections

33.1 Maskinterp: : — :, 10 min

- Code is in class directory
- Package looks for bad pixels in mask. For each bad pixel, it:
 - Makes an aperture centered on the bad pixel
 - Tests whether there are enough good pixels to fit
 - Tests for weighting on one side or the other
 - Fits surface model to good pixels in aperture
 - Evaluates function at bad pixel
 - Updates *copy* of data
 - Does not update mask!
 - Those data are not real
 - Those data are not used in later interpolations
- See `testinterp.pro` and try doing some stuff
- Use Gaussian if all sources are stars, else use 2nd-order polynomial
- Example, including syntax
- Good idea to write wrapper that runs it and does header entries

33.2 Maskinterp Demo: : — :, 5 min

33.3 General Linear Fitting: : — :, 10+5 min

- First example extends the line fit to linear scaling of arbitrary models

- This is a quick-and-dirty method, useful because everyone has a linfit routine
- Take each prediction of a nominal model,
- Multiply by a constant,
- Add another constant
- 2 free parameters, like a line fit
- Can use linear fitting routine to find parameters
 - Calculate a “nominal” model
 - Treat x as the parameter in a parametric equation
 - Calculate model for x values in data
 - Pair those with y values in data
 - Plot $d = y_{data}$ vs. $m = y_{model}$
 - No error in m , but error in d
 - So, d goes on vertical axis
 - Fit a line!
 - Slope is how much you multiply model by
 - Intercept is how much you add to model
 - If model has just multiply parameter, intercept is used as a final background fit
- Can still do if data are in more than 2 axes
- E.g., model depends on x and y in an image
- 06fitting

Chapter 34

Periodic Signals and Power Spectra

STOP! Replace with fitting/MCMC. About 250 minutes of lecture.

Week 10.

34.1 Check In: : — :, 10 min

- Questions?

34.2 What is a Power Spectrum?: : — :, 5 min

- Power spectrum evaluates how periodic a function is
- Measures amplitudes of oscillation at various frequencies
- A function $h(t)$ is in the *time domain*, intervals in units of time
- Its power spectrum is in the *frequency domain*, intervals in units of 1/time
- Spatial frequencies are frequencies too! Units of distance and 1/distance
 - Frequency: cycles per sec
 - Angular frequency: radians per sec
 - Wavenumber: cycles per meter
 - Planetary wavenumber: wave periods per circumference at a given latitude
- We'll do the simple, 1D case
- Others derived from it

34.3 Goals of Calculating the Power Spectrum: : — :, 5 min

- Interested in power spectrum for itself
 - Want $P(\nu)$
 - “Harmonics” of sound, **spectrum of light!**
 - Seek peak to guide periodogram analysis
 - If frequencies communicate, cascade theories predict PS
 - Noise has a power spectrum! Flat = “white”, sloped = “red”
- Convolutions easier to calculate in frequency domain: mathematical convenience
- Astronomy applications:
 - Exoplanetary radial velocity detection method
 - Estimate rotation period from lightcurve (pulsar, asteroid, planet, moon)
 - Finding and analyzing planetary waves

34.4 Orthonormal Basis Functions: : — :, 10 min

- Decompose an arbitrary function $h(t)$ into its amplitudes in a set of orthonormal basis functions $b_j(t)$

$$h(t) = \sum_{j=0}^n A_j b_j(t) \quad (34.1)$$

- A_j is the j th coefficient (amplitude) (of basis function $b_j(t)$)
- Basis function $b_j(t)$ could be $\sin(jt)$, Hermitian polynomials, Chebychev polynomials, etc.
- Useful if the functions have properties that make them easy to calculate with
- Orthonormal basis functions must be *orthogonal* functions:

$$0 = \int_{-\infty}^{\infty} b_j(t) b_k(t) dt, j \neq k \quad (34.2)$$

- Any even function is orthogonal to any odd function
- Show that $\sin(jt)$ are orthogonal for integer j on board
- Need both sin and cos, or can only make an even or odd function
- That is, need a *complete* set of functions
- The coefficient of a particular basis function, $b_j(t)$:

$$A_j = \int_{-\infty}^{\infty} h(t) b_j(t) dt \quad (34.3)$$

34.5 Transforms: : — :, 5 min

- Make the coefficient a continuous function
- Make the basis functions a continuous set, like $e^{i\omega t}$
- Do one integral:

$$H(f) = \int_{-\infty}^{\infty} h(t)b(f, t)dt \quad (34.4)$$

- $H(f)$ and $h(t)$ are a *transform pair*

34.6 Sound Example: : — :, 10 min

- Want amplitude vs. pitch: power spectrum
 - Air pressure is $h(t)$, a continuous function
 - Do transform with oscillation as basis function
 - $H(f)$ is amplitude vs. pitch
- Consider how the ear hears: hairs of different natural frequencies
- We know oscillations are sinusoidal, but what about phase shifts?
- Break sound into sin and cos basis functions, with S_j and C_j as amplitudes
- Pair the S_j and C_j for sines and cosines with the same argument
- $\sum C_j \cos(f_j t) + S_j \sin(f_j t)$
- Use S_j and C_j as the real and imaginary components of a complex vector
- The vector has amplitude $a = \sqrt{S_j^2 + C_j^2}$ and angle $\phi_0 = \arctan(S_j/C_j)$
- Since $e^{i\theta} = \cos \theta + i \sin \theta$,
- To evolve it in time, $ae^{ift}e^{i\phi_0} = ae^{i(ft+\phi_0)}$
- Or, make a complex: ae^{ift}
- This includes both sin and cos
- Fourier Transform:

$$H(f) = \int_{-\infty}^{\infty} h(t)e^{2\pi ift} dt \quad (34.5)$$

$$h(t) = \int_{-\infty}^{\infty} H(f)e^{-2\pi ift} df \quad (34.6)$$

- We can also work in angular frequency:

$$H(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} h(t)e^{2\pi i\omega t} dt \quad (34.7)$$

$$h(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} H(\omega)e^{-2\pi i\omega t} d\omega \quad (34.8)$$

34.7 Properties: : — :, 10 min

- $h(t)$ and $H(f)$ are in general complex
- The real and imaginary components of $H(f)$ can be converted to amplitude and phase
- If $h(t)$ is real, $H(f) = [H(-f)]^*$
- There are many other properties, see, e.g., *Numerical Recipes* section 12.0
- Convolution: $g(t) \circ h(t) = \text{FT}^{-1}[G(f)H(f)]$
- This allows deconvolution! Just transform and divide.
- $\text{FT}(a + b) = \text{FT}(a) + \text{FT}(b)$
- This allows analysis of noise separately from signal, if $a + b = \text{signal} + \text{noise}$
- $H(0)$ is the mean of the data, the DC offset
- H is the amplitude spectrum
- The power spectrum is $[\text{abs}(H)]^2$
- Literally, how much energy flowing per unit time at that frequency
- The power spectral density is $\frac{|H|^2}{H(0)^2}$

34.8 Real Data, Sampling, and Aliasing: : — :, 10 min

- Fourier transform (FT) is defined continuously from $-\infty < t < \infty$
- Real data covers finite intervals
- Real data is sampled, not continuous
- Real data may not be sampled evenly in time
- If sampling interval is Δt then $h_n = h(n\Delta t)$, $n = \dots - 3, -2, -1, 0, 1, 2, 3\dots$
- You must sample a sine wave at least twice per cycle to distinguish it from, say, a constant

- Diagram $\sin(x)$ and $y = 1$, sampled once and twice per cycle
- The minimal sampling (Nyquist) frequency is $f_c = \frac{1}{2\Delta t}$
- Often work in units of Δt ; then $f_c=1/2$
- Good news: if $f_{\max} < f_c$, then signal is *bandwidth limited*
- Information content of bandwidth-limited signals is completely determined by h_n !
- If signal is not substantially bandwidth-limited, power in power spectrum *aliases* from $f < -f_c$ and $f > f_c$ into $f_c < f < f_c$
- Can't tell for sure if a spectrum contains aliased power
- But, if spectrum is not approaching zero at $\pm f_c$, it's probably aliased
- Often there are natural limits
- E.g., atmospheric seeing for an image: PSF limits spatial wavelength of signal

34.9 Fourier Transform: : — :, 10 min

- The continuous (analytical) Fourier transform

$$H(f) = \int_{-\infty}^{\infty} h(t)e^{2\pi ift} dt \quad (34.9)$$

$$h(t) = \int_{-\infty}^{\infty} H(f)e^{-2\pi ift} df \quad (34.10)$$

If we define

$$\omega = 2\pi f, \quad (34.11)$$

then

$$H(\omega) = \int_{-\infty}^{\infty} h(t)e^{i\omega t} dt \quad (34.12)$$

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(\omega)e^{-i\omega t} d\omega \quad (34.13)$$

Often you'll see a factor if $\frac{1}{\sqrt{2\pi}}$ in front of both formulae involving ω .

34.10 Discrete Fourier Transform: : — :, 10 min

- Say we have N evenly-sampled values of $h(t)$
- $h_k = h(t_k), t_k = k\Delta t, k = 0, 1, 2, 3, \dots$
- If $h(t) = 0$ outside a range, cover the range with the N points
- Otherwise, make sure the range covered is typical of $h(t)$
- We can only get N pieces of information from N inputs
- Sample at $f_n = \frac{n}{N\Delta t}, n = -\frac{N}{2}, \dots, \frac{N}{2}$, min and max frequency are Nyquist
- Discrete Fourier Transform is:

$$H(f_n) = \int_{-\infty}^{\infty} h(t)e^{2\pi i f_n t} dt \approx \sum_{k=0}^{N-1} h_k e^{2\pi i f_n t_k} \Delta t = \Delta t \sum_{k=0}^{N-1} h_k e^{2\pi i k n / N} \quad (34.14)$$

$$H_n = \sum_{k=0}^{N-1} h_k e^{2\pi i k n / N} \quad (34.15)$$

- Here, we've gotten rid of the pesky Δt , so $H(f_n) = \Delta t H_n$
- One more convention makes things easier
- It's clear that this assumes the data are periodic with period N
- Rather than having $f = 0$ be at $n = N/2$, have it at $n = 0$
- Then $0 < f < f_c$ runs over $1 \leq n \leq N/2 - 1$
- $-f_c < f < 0$ runs over $N/2 + 1 \leq n \leq N - 1$
- $n = N/2$ is $f_c = -f_c$
- Chart this
- The inverse transform is

$$h_k = \frac{1}{N} \sum_{n=0}^{N-1} H_n e^{-2\pi i k n / N} \quad (34.16)$$

34.11 DFT Speed: : — :, 5 min

- How much computing time does that take?
- If $W = e^{2\pi i / N}$ then $H_n = \sum W^{nk} h_k$
- That's a matrix multiplication, naively requiring $O(N^2)$ computations
- There's a faster algorithm that does it in $N \log_2 N$ time!
- The algorithm has been "discovered" at least a dozen times, starting with Gauss in 1805!

34.12 Amplitude, Phase, Power Spectrum in Python: : — :, 50 min

- `fft-demo.py` - save waveforms and harmonics until later
- Steps for FFT:
- Variables are $(t) \Leftrightarrow (f)$
- Times and frequencies of the elements of t and f are implicit, even spacings
- Read data into array t (can be complex)
- `f=np.fft.fft(t)`
- `t=np.fft.ifft(f)`
- Power: `P=np.abs(f)**2`
- Power Spectral Density: `PSD=P/P[0]`
- Chance that Poisson process (e.g., photons) produces PSD F by chance is $p(F) = e^{-F}$
- A 3σ detection thus requires `PSD ~6.5`
- Amplitude: `a=np.abs(f)`
- Phase: `ph=np.arctan2(np.imag(f), np.real(f))`
- Frequencies: `nu = np.arange(N/2)/(N*dt)`
- Plot: `plt.plot(nu, P[0:N/2-1])`
- Signals and noise

34.13 FFT Algorithm: : — :, 10 min

- A DFT of length N is the sum of 2 DFTs of length $N/2$
- One of these is from the even-numbered n , the other from the odd:

$$H_n = \sum_{k=0}^{N-1} e^{2\pi i k n / N} h_k \quad (34.17)$$

$$= \sum_{k=0}^{N/2-1} e^{2\pi i n (2k) / N} h_{2k} + \sum_{k=0}^{N/2-1} e^{2\pi i n (2k+1) / N} h_{2k+1} \quad (34.18)$$

$$= \sum_{k=0}^{N/2-1} e^{2\pi i n (2k) / N} h_{2k} + W^n \sum_{k=0}^{N/2-1} e^{2\pi i n (2k) / N} h_{2k+1} \quad (34.19)$$

$$= H_n^e + W^n H_n^o \quad (34.20)$$

$$W = e^{2\pi i / N} = (-1)^{2/N} \quad (34.21)$$

- W is a constant, complex, unit vector with $\theta = 2\pi/N$
- Raise that to $n = 0 \dots N - 1$ and it rotates once in N samples
- H_n^e is the DFT of length $N/2$ from the even h_k s
- H_n^o is from the odd ones
- n still runs $0, \dots, N - 1$, so H_n^e and H_n^o span the interval but take half the data
- But, you can compute H_n^e and H_n^o using the same method, giving H_n^{oe} and H_n^{ee}
- Each of these now spans the interval but samples 1/4 of the data
- ...and again, until there is $H_n^{oooooooooooo} = h_n$
- 1-point transforms are identities: plug in $N = 1$ above
- The splits make a binary tree of sums
- Which value of n corresponds to $oooooooooooo$?
- We got $oooooooooooo$ by successive subdivisions of the data
- The first e/o decision was like the 1 bit in a binary number
- The second was like the 2 bit, and so on
- Replace $e \rightarrow 0$ and $o \rightarrow 1$
- Reverse the bit order
- This is the binary representation of n !
- Re-order h_n according to new n (it's just swaps)
- Do each set of sums in binary tree to reconstruct H_n in steps
- If you don't have 2^m elements, pad with 0 to next 2^m boundary

34.14 FFT Speed: : — :, 5 min

- Compute time: each h_n gets summed once, but there are only $\log_2 N$ binary splits
- As long as the binary reordering is quick, there are N F 's to calculate, each taking $\log N$ time
- FFT is an $N \log_2 N$ algorithm
- On a 1 MHz CPU, the time to calculate a 10^6 -point DFT is 1 week
- With FFT, it's 30 sec!

Chapter 35

Fourier Transform Applications

35.1 Check In: : — :, 10 min

- Questions?

35.2 HW11 Peer Evaluation: : — :, 45 min

35.3 Waveforms and Harmonics: : — :, 10 min

- Say a function $h(t)$ is periodic with frequency f `draw`
- A *periodic waveform* is repeated exactly to make $h(t)$
- Contributing Fourier components must also be periodic with that period
- Any that aren't will produce signal that is not the same from one period to the next `draw`
- $\sin(x)$ is periodic with frequency $1/2\pi = f$
- $\sin(nx)$ is *also* periodic with period $1/2\pi = f$, (and of course with period $n/2\pi = n.f$)
- The FT of $h(t)$ will have a strong response at frequency f . This is the *fundamental*.
- It will have power, possibly stronger, at *harmonic* frequencies $2f, 3f, 4f, \dots$
- So, the FFT will have a sequence of evenly-spaced spikes, starting at the fundamental
- `fft-demo.py`

35.4 Periodograms: : — :, 15 min

- So, what about the period?
- We want a periodogram: a plot of a given, precise width where the data are repeated

- In Python:
 - have arrays `h` and `t`
 - `pert = t % period`
 - `plt.plot(pert, h, '.')` (or something that doesn't connect points)
 - `lect23-demo.py`
- Problem: what if we don't know the period?
- (Redo demo with wrong period)
- If our period puts just ± 1 extra periods into the span of the data, it looks random
- We could guess
- We could subsample the data
- We could do some easy measurement of the period, and fine-tune that
- Some pulsar datasets have 2^{21} bins, or about 2 million measurements
- Each bin is 0.00064 seconds long
- We think the pulsar goes at about 30 Hz
- That's $2 \times 10^6 \times 0.00064 \text{ sec} \approx 1200 \text{ sec}$
- $1200 \text{ sec} \times 30 \text{ cycles / sec} = 36,000 \text{ cycles}$
- So, we need a period estimate good to within 1 part in 36,000 before we start with the periodogram!
- Then, bin the periodogram in time
- Calculate median of each bin
- Calculate variance around median per bin
- Calculate total variance
- Vary period slightly to minimize variance
- Done when variation small enough to make you happy, or no improvement

35.5 Estimating a Period With the FFT: : — :, 10 min

- Know the FT of periodic signal has evenly-spaced spikes with zeros between
- Know the first spike (not strongest) is fundamental
- Know remaining spikes are at integer multiples of first spike frequency
- Measure center frequency of each spike, and uncertainty
- Non-integer, must interpolate
- Divide that frequency and uncertainty by harmonic number
- “Optimally combine” = average weighted by uncertainties (Bevington!)
- This estimates the frequency, and therefore the period, quite accurately
- Now finish off with periodogram

35.6 Wavelets: : — :, 45 min

- ast5765-wavelets.ppt
- `wavelet-demo.pro` FINDME: convert to Python!

***** END OF STUFF TO CUT *****