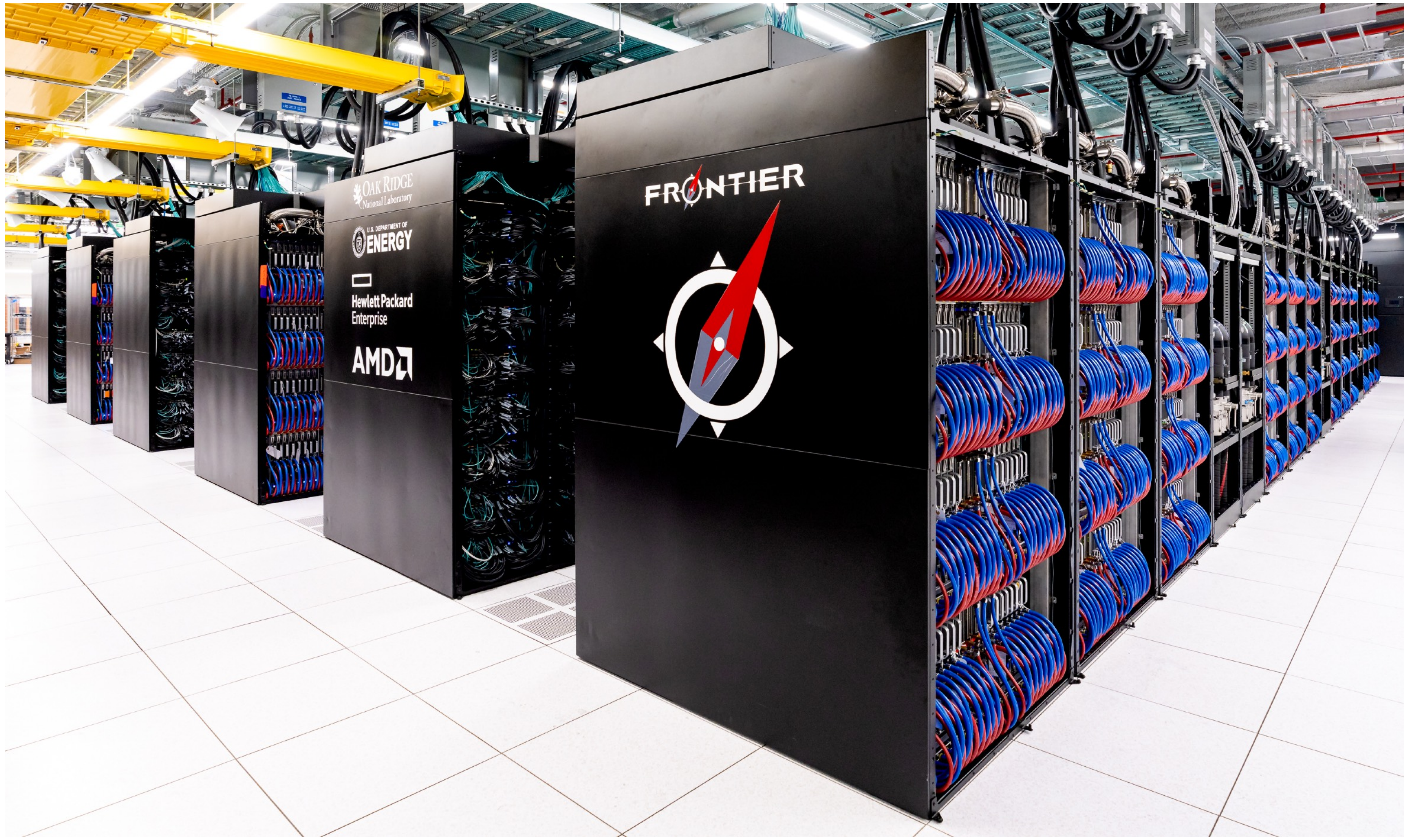
Computing in Astronomy



Brian O'Shea (oshea@msu.edu)
Astro 1st year PhD student seminar
Oct. 7, 2025

Slides at
https://bit.ly/MSUAstroComp
(or QR code)

What are we talking about today?

1. An overview of how computing is used in astronomy & astrophysics
2. Specific computing resources at MSU and nationally
3. Tools and skills that are particularly important for you to know, both for grad school and your future career
4. Resources for growing your computing skillset
5. Courses relating to computing that are available to you at MSU
6. The CMSE graduate certificates and dual PhD, and why those might be interesting/useful
7. Anything else that you might be interested in!

Who am I?

- At MSU since 2008
- Professor in CMSE, P&A, FRIB
- Director of ICER and the Bioinformatics Core
- Computational astrophysicist
  - Galaxy evolution, fluid dynamics, plasma physics
  - Open source code development
  - Codes and algorithms for exascale/next-gen computing
- I teach the grad Computational Astrophysics course (PHY 905 special topics) every 2 or 3 years - next time will be next semester (Spring 2025), Tu/Th morning @ 10:20 a.m. in 1300 BPS.

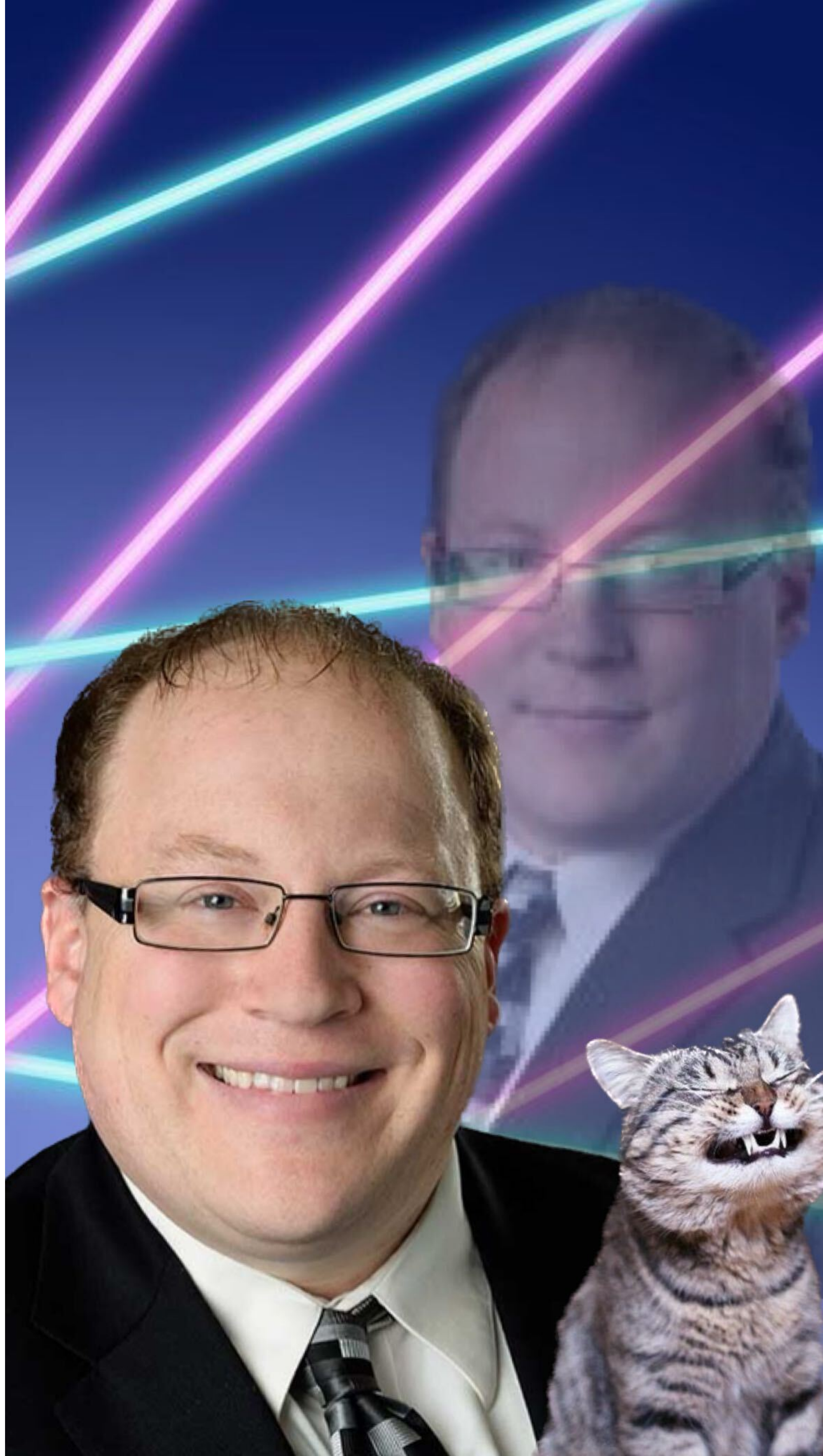


Photo c/o MSU Computational Education Research Lab

Why is it important to develop computing skills if you are pursuing a PhD in Astronomy*?

- Astronomy is deeply quantitative and computational at nearly every stage of the research process - from data collection, to analysis, to modeling.
- Telescopes and simulations generate huge and complicated datasets, and you need to be able to clean, process, and analyze that data, use statistical/ML techniques to identify patterns or rare phenomena, and work with standard data formats, datasets, etc.
- Both astronomy and theoretical astrophysics heavily relies on numerical modeling and simulations for interpreting data, understanding error of various sorts, making predictions, etc.
- You can write scripts to automate repetitive tasks, create reproducible data analysis pipelines, and you can use version control to track what you do and collaborate.
- Computing skills are **highly transferable beyond academia**! Many graduates from our PhD program (and those like it) go into data science, software engineering, finance, engineering, or related disciplines, and these skills are a huge professional asset.

* the "and theoretical Astrophysics" is implicit here and elsewhere in this talk!

MSU computing resources

- The Institute for Cyber-Enabled Research at MSU (ICER; https://icer.msu.edu/)
  - Local CPU+GPU computing clusters with both command-line and web access (https://ondemand.hpcc.msu.edu/), including data-intensive computing resources (the Data Machine)
  - Workshops (in person and asynchronous via Desire2Learn) and courses
  - Very friendly Research Consultants (some with astro backgrounds!) to help you find the right tools, optimize your workflows, and solve problems.
  - Cloud Computing Fellowship and (new!) Open Science Fellowship - targeted at PhD students, to complement your research!
  - All MSU researchers can get access to ICER resources for free - faculty need to create allocations and can add you for compute!

National computing resources

- NSF ACCESS (https://access-ci.org/)
  - Ecosystem of NSF-funded computing resources - CPU, GPU, cloud computing, all sorts of things!
  - Lots of events and training, many are virtual, some remotely hosted by ICER (https://support.access-ci.org/events)
  - Resources are available for free - allocation application required (of varying sizes depending on request), credits given so you can experiment on different systems!
- Open Science Grid (https://osg-htc.org/) - high throughput computing (for “pleasantly parallel” problems, but lots of resources available)
- NASA, DOE have their own resources but they are less easy to obtain!

Important tools and skills

- Most important:
    - Understanding file systems, local vs. remote vs. cloud storage (important for organization, code development)
    - Working at the Linux command line (including ssh and scp)
    - Bash/shell scripting - for automating workflows and managing files on Linux system
    - Python (dominant programming language for data analysis, modeling, visualization), esp. Astropy, NumPy, SciPy, pandas, matplotlib, Jupyter notebooks
    - Git (using GitHub or GitLab) - for version control and collaboration
    - Skills: writing modular code, debugging and profiling code, documenting your work (Markdown!)

Important tools and skills

- Highly useful:
  - Working with large datasets - working with astronomical data formats, cleaning data, managing data, and databases (e.g., online databases such as Vizier or MAST)
  - Using high performance computing systems to <u>accelerate</u> your work and for working with <u>massive datasets</u> - managing and submitting batch jobs, managing dependencies and environments (e.g., Conda, Docker), using parallelization tools (MPI, OpenMP, Python multiprocessing), using automation tools (bash scripting, Snakemake)
  - Programming with a compiled language (C++ is most useful) for writing high-performance code.
  - Skills:
    - thoughtful data visualization and communication (including interactive visualizations!)
    - Numerical methods for modeling (linear algebra, solving differential equations) - important to understand what these do!

Important tools and skills

- Advanced/emerging methods
  - Machine learning and AI (deep learning, generative AI): scikit-learn, TensorFlow, PyTorch, various LLMs
  - Bayesian inference and Markov Chain Monte Carlo methods (and statistical sampling/modeling methods more generally)
  - Databases and SQL (working with large catalogs)
  - GPU computing (for high performance computing, neural networks/deep learning/generative AI)

Resources for growing your computational skillset

- ICER training
  - https://icer.msu.edu/education-and-events
  - Every fall: ICER courses on “HPC with Python” and “Reproducible Computational Workflows” as CMSE 890 special topics courses.
- ACCESS training: https://support.access-ci.org/events
- Software Carpentry (linux, Git, reproducible workflows): https://software-carpentry.org/lessons/ (MSU also hosts Software Carpentries workshops, typically through the Library+ICER+Digital Humanities - often during spring break)
- Lots and lots of MSU courses (we’ll address that momentarily)

MSU courses relating to computing

- CMSE 801*: Intro to Computational Modeling and Data Analysis (Python; every semester; *does not count for AST or CMSE PhD program grad elective)
- CMSE 802*: Methods in Computational Modeling (Python; most semesters; *does not count for AST or CMSE PhD program grad elective)
- CMSE 820: Mathematical Foundations of Data Science (Matlab; spring; very proof-heavy)
- CMSE 821, 823: Numerical Differential Equations, Numerical Linear Algebra (Python/Matlab; both in the fall)
- CMSE 822: Parallel Computing (C++; in the spring)
- CMSE 402: Data Visualization Principles and Techniques (Python; Spring of even years; sometimes more frequently)
- CMSE 404: Intro to Machine Learning (Python; every semester)
- PHY 905 (section varies): Computational Astrophysics and Astrostatistics (Python; Spring roughly every other year, **including Spring 2026**)

Note: (1) you can take classes outside of PA and CMSE!
(2) some classes are more applied math than practical computing!
(3) you may need to request permission to take some classes - look at catalog early!

MSU courses relating to computing

- CSE 440: Intro to Artificial Intelligence (Language varies; every semester)
- CSE 482: Big Data Analysis (Language varies; every spring)
- CSE 840: Computational Foundations in AI (every fall)
- CSE 841: Artificial Intelligence (every spring)
- CSE 847: Machine Learning (every spring)
- CSE 849: Deep Learning (every spring)
- CSE 881: Data Mining (every spring)
- CMSE and CSE special topics courses (CMSE 491, 492, 890, CSE 491, 492, 891, 895, 941)

Note: (1) you can take classes outside of PA and CMSE!
(2) some classes are more applied math than practical computing!
(3) you may need to request permission to take some classes - look at catalog early!

CMSE graduate certificates and dual PhD

- CMSE Graduate Certificates: 9 credits (3 courses); appear on your transcript
  - Computational Modeling: 2 core CMSE courses (801, 802, 820-823), plus one additional computational course from a list (or via CMSE grad director approval).
  - High Performance Computing: CMSE 822 (Parallel Computing), any other 800-level CMSE course except CMSE 801, 802 or select other 800, 900-level courses (400-level is OK if 8/900-level is unavailable)
- Dual PhD with CMSE (Astrophysics and CMSE in this case)
  - Dissertation must blend astronomy+astrophysics and computational+data science in a significant way
  - Courses are a hybrid of Astrophysics and CMSE PhD program requirements (3 AST core; 2 CMSE core)
  - PhD committee is a mix of PHY/AST/CMSE faculty

More info at https://cmse.msu.edu/Academics/grad_program/index.aspx (or just ask me)

Any questions?