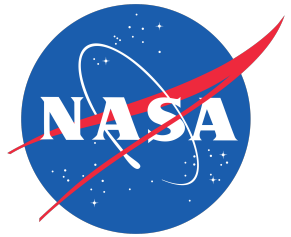
Designing and Implementing a Computational Methods Course for Upper-level Undergraduates and Postgraduates in Atmospheric and Oceanic Sciences

Ethan L. Nelson, Tristan S. L'Ecuyer,
Alyson Douglas, Zachary Hansen

University of Wisconsin–Madison

December 13, 2017

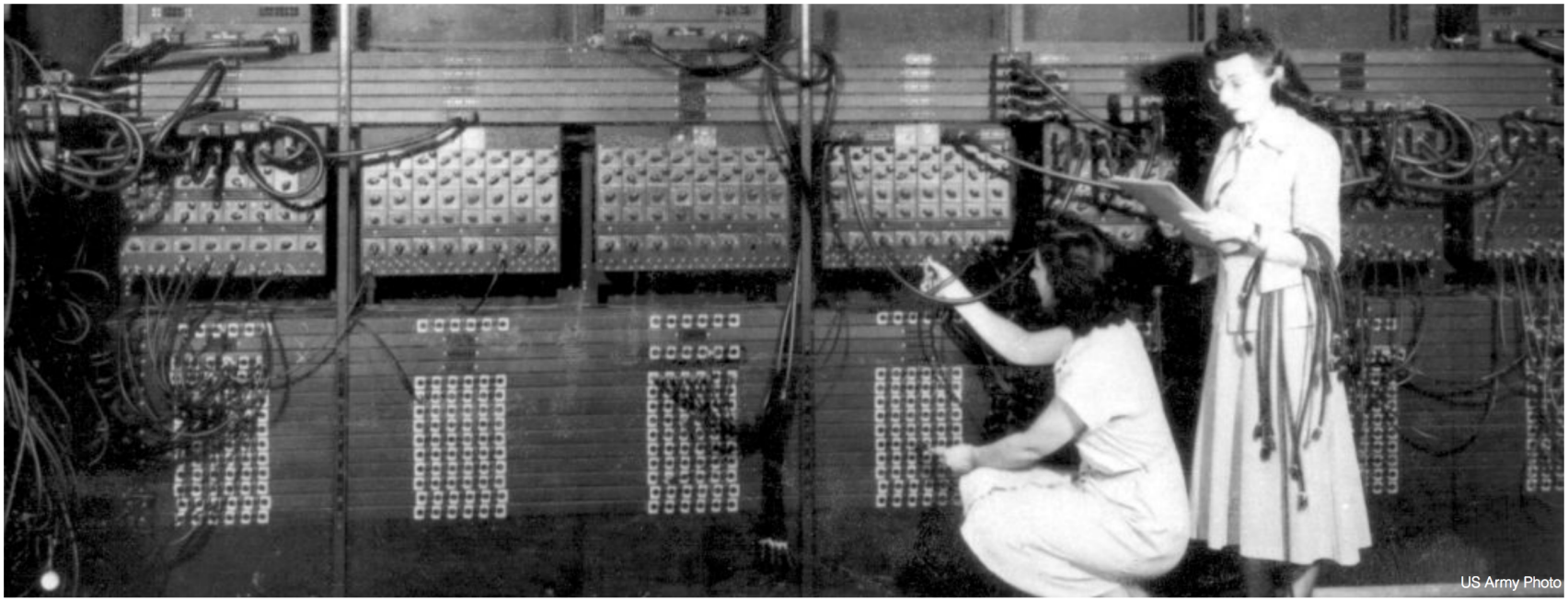


Motivations

- Students entering research or upper-level courses were often not equipped with programming experience.
- Generic programming courses usually focused on less useful languages for the field.
- Growing trends include reproducible research, utilization of open source technologies, and collaborative development.

Computational Methods in AOS

Assignments Lectures Schedule



Course Homepage

Welcome to the homepage of *AOS 573: Computational Methods in Atmospheric and Oceanic Sciences* at the University of Wisconsin–Madison for the Spring 2017 semester. Here you will find course information including the syllabus and schedule, lecture information, and lab assignments.

This class is designed to provide you practical programming experience in three languages used commonly in the field of Atmospheric and Oceanic Sciences, while completing lab exercises and projects aimed toward discipline-specific applications.

Computational Methods in AOS

Computational Methods in Atmospheric and Oceanic Sciences

Course Attributes

Discipline specific

Course Attributes

Discipline specific

Project focused

Course Layout

Computational Methods in AOS

Assignments Lectures Schedule

Schedule

Week	Topic	Lecturer	Language	Lecture	Assignment
1	Introduction to Programming	Ethan	Bash, Git		
2	Programs, printing, reading, loops, subprograms	Ethan	Fortran		
3	Multidimensional arrays and more I/O	Ethan	Fortran		
4	Fortran CLI and Modules	Ethan	Fortran		
5	Fortran Modeling	Feiyu	Fortran		
6	Introduction to Matlab	Zach	Matlab		
7	Matlab Data I/O and Functions	Zach	Matlab		
8	Statistics	Zach	Matlab		
9	Animations and Movies	Zach	Matlab		
10	Introduction to Python	Alyson	Python		
11	Input and Output, Binning	Alyson	Python		
12	Plotting Routines and Graphics	Alyson	Python		
13	Advanced Plotting	Alyson	Python		
14	Releasing You into the Wild	Ethan	Python and More		

Course Layout

Computational Methods in AOS

Assignments Lectures Schedule

Schedule

Intro <

Week	Topic	Lecturer	Language	Lecture	Assignment
1	Introduction to Programming	Ethan	Bash, Git		
2	Programs, printing, reading, loops, subprograms	Ethan	Fortran		
3	Multidimensional arrays and more I/O	Ethan	Fortran		
4	Fortran CLI and Modules	Ethan	Fortran		
5	Fortran Modeling	Feiyu	Fortran		
6	Introduction to Matlab	Zach	Matlab		
7	Matlab Data I/O and Functions	Zach	Matlab		
8	Statistics	Zach	Matlab		
9	Animations and Movies	Zach	Matlab		
10	Introduction to Python	Alyson	Python		
11	Input and Output, Binning	Alyson	Python		
12	Plotting Routines and Graphics	Alyson	Python		
13	Advanced Plotting	Alyson	Python		
14	Releasing You into the Wild	Ethan	Python and More		

Course Layout

Computational Methods in AOS

Assignments Lectures Schedule

Schedule

Intro

Fortran

Week	Topic	Lecturer	Language	Lecture	Assignment
1	Introduction to Programming	Ethan	Bash, Git		
2	Programs, printing, reading, loops, subprograms	Ethan	Fortran		
3	Multidimensional arrays and more I/O	Ethan	Fortran		
4	Fortran CLI and Modules	Ethan	Fortran		
5	Fortran Modeling	Feiyu	Fortran		
6	Introduction to Matlab	Zach	Matlab		
7	Matlab Data I/O and Functions	Zach	Matlab		
8	Statistics	Zach	Matlab		
9	Animations and Movies	Zach	Matlab		
10	Introduction to Python	Alyson	Python		
11	Input and Output, Binning	Alyson	Python		
12	Plotting Routines and Graphics	Alyson	Python		
13	Advanced Plotting	Alyson	Python		
14	Releasing You into the Wild	Ethan	Python and More		

Course Layout

Computational Methods in AOS

Assignments Lectures Schedule

Schedule

Intro

Fortran

Matlab

Week	Topic	Lecturer	Language	Lecture	Assignment
1	Introduction to Programming	Ethan	Bash, Git		
2	Programs, printing, reading, loops, subprograms	Ethan	Fortran		
3	Multidimensional arrays and more I/O	Ethan	Fortran		
4	Fortran CLI and Modules	Ethan	Fortran		
5	Fortran Modeling	Feiyu	Fortran		
6	Introduction to Matlab	Zach	Matlab		
7	Matlab Data I/O and Functions	Zach	Matlab		
8	Statistics	Zach	Matlab		
9	Animations and Movies	Zach	Matlab		
10	Introduction to Python	Alyson	Python		
11	Input and Output, Binning	Alyson	Python		
12	Plotting Routines and Graphics	Alyson	Python		
13	Advanced Plotting	Alyson	Python		
14	Releasing You into the Wild	Ethan	Python and More		

Course Layout

Computational Methods in AOS

Assignments Lectures Schedule

Schedule

Week	Topic	Lecturer	Language	Lecture	Assignment
1	Introduction to Programming	Ethan	Bash, Git		
2	Programs, printing, reading, loops, subprograms	Ethan	Fortran		
3	Multidimensional arrays and more I/O	Ethan	Fortran		
4	Fortran CLI and Modules	Ethan	Fortran		
5	Fortran Modeling	Feiyu	Fortran		
6	Introduction to Matlab	Zach	Matlab		
7	Matlab Data I/O and Functions	Zach	Matlab		
8	Statistics	Zach	Matlab		
9	Animations and Movies	Zach	Matlab		
10	Introduction to Python	Alyson	Python		
11	Input and Output, Binning	Alyson	Python		
12	Plotting Routines and Graphics	Alyson	Python		
13	Advanced Plotting	Alyson	Python		
14	Releasing You into the Wild	Ethan	Python and More		

Intro

Fortran

Matlab

Python

Course Layout

Computational Methods in AOS

Assignments Lectures Schedule

Schedule

Week	Topic	Lecturer	Language	Lecture	Assignment
1	Introduction to Programming	Ethan	Bash, Git		
2	Programs, printing, reading, loops, subprograms	Ethan	Fortran		
3	Multidimensional arrays and more I/O	Ethan	Fortran		
4	Fortran CLI and Modules	Ethan	Fortran		
5	Fortran Modeling	Feiyu	Fortran		
6	Introduction to Matlab	Zach	Matlab		
7	Matlab Data I/O and Functions	Zach	Matlab		
8	Statistics	Zach	Matlab		
9	Animations and Movies	Zach	Matlab		
10	Introduction to Python	Alyson	Python		
11	Input and Output, Binning	Alyson	Python		
12	Plotting Routines and Graphics	Alyson	Python		
13	Advanced Plotting	Alyson	Python		
14	Releasing You into the Wild	Ethan	Python and More		

Intro: Week 1
Fortran: Weeks 2–5
Matlab: Weeks 6–9
Python: Weeks 10–13
Outro: Week 14

Concepts Conveyed

- Programming competency is not tied to one language; it is about solving a problem.
- Programming languages differ in the syntax they use.
- Language-agnostic concepts sprinkled throughout the course:
  - Conceptual code models
  - Data wrangling
  - Practices for reproducibility and collaboration

Course Delivery

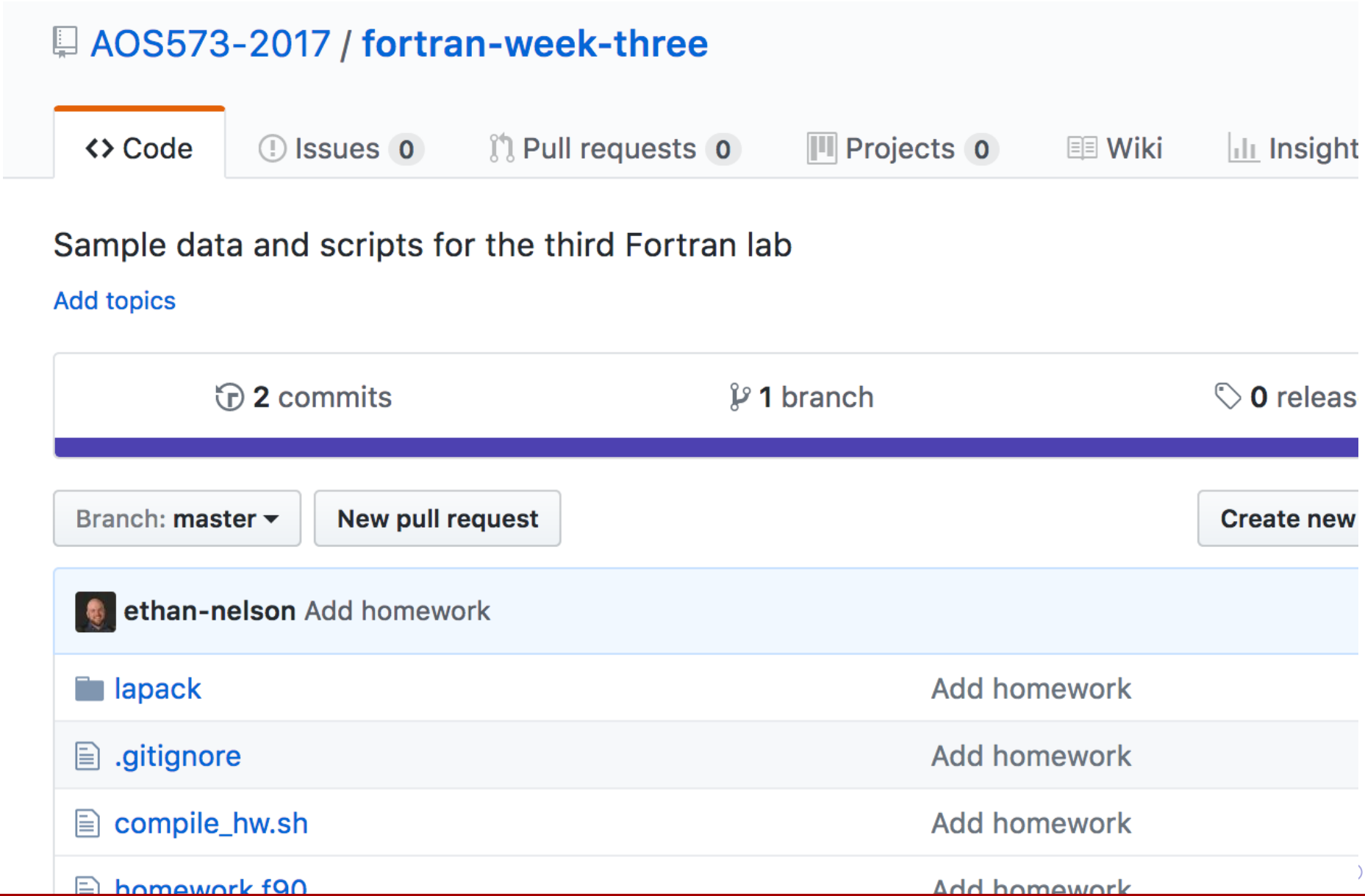
- Half "lecture", half laboratory work.
- Lecture was teaching/learning while coding.
  - Provided real-time feedback and ability to tinker.
  - Transference of tacit knowledge.
- Long-form notes posted online for reference.
- Sample scripts and data downloaded via GitHub set foundation.
- Content built up over course of a language to solve a final problem.

Example Research Problem

Question: *How does Lake Michigan affect conditions in Milwaukee?*

1. Build a Fortran submodule for vector decomposition and unit conversion.
2. Define type of and read in text data; write out formatted data
3. Mask data and read in input from command line
4. Compute relative humidity in Milwaukee composited with different wind conditions.

Example Week Repository



Example Long-Form Notes

Allocatable Variables

So far we have been *hard-coding* the shapes of variables, which means we manually specify the lengths of each dimension prior to run time. It may be the case, however, that we will not know the length of each dimension needed until run time. This behavior could occur when we are reading in files with different sized variables. Or we could be filling a variable only with variables that meet a certain criteria. These cases require using a *dynamic allocation* of variable shape. Dynamic allocation punts the reservation of memory for a variable until we specifically choose to allocate it and it also provides us the ability to free (or deallocate) that memory once we are done with the variable.

In Fortran, dynamic allocation is defined with the `ALLOCATABLE` keyword (similar to specifying `PARAMETER`) and using colons to specify the number of dimensions of a variable. The shape is defined using the `ALLOCATE` statement and the variable memory can be discarded using the `DEALLOCATE` statement. If you are working with a large amount of data that is read in and then manipulated or transferred into a few other variables, it may be necessary to use dynamic allocation if for nothing else than to ensure your program is able to run with your machine's memory.

Here we will define an allocatable variable of three dimensions as well as three variables that we will later use to allocate the lengths of each dimension:

```
PROGRAM arrays
! by Bucky Badger
! This program will work with some arrays in Fortran
IMPLICIT NONE

REAL :: test(3,2)
CHARACTER :: words(5)
CHARACTER(34) :: newwords

REAL, ALLOCATABLE :: modeldata(:,:,:)
INTEGER :: nlon, nlat, ntime

nlon = 360
nlat = 180
ntime = 24 * 365

PRINT *, ALLOCATED(modeldata)
PRINT *, SHAPE(modeldata)
```

Successes

- Students were very receptive to the guided learning and lab session.
- Slower pace set by typing during lecture was preferred.
- Students successfully completed projects that involved utilizing conceptual models, data management, and language-agnostic skills.
- Release of lecture materials online will (hopefully) assist in collaborating with other universities and for future iterations of the course.

Challenges

- Combination of undergraduate and graduate students was complicating at times.
- Pair programming during lab session had mixed results.
- Some spin-up time was required with this lecture format for instructors.
- Three languages may be too many–possibly focus on a compiled and a non-compiled.

Closing

- New discipline-specific computational methods provided an opportunity for students to become familiar with programming in an applied manner.
- Broad concepts were covered such as data manipulation, modularized and documented code, and reading documentation.
- Active learning by doing was appreciated by students.

Thanks!

https://aos573.github.io

Thanks to NASA HQ under the NASA Earth and Space Science Fellowship Program (#NNX14AL35H) for the freedom to carry out this course concept.