# Table of Contents

# Computational instruction with Jupyter notebooks

Philip Austin, Department of Earth, Ocean and Atmospheric Sciences, UBC

Notebook available at https://github.com/phaustin/mcgill_2019_talk (https://github.com/phaustin/mcgill_2019_talk)

# Part 1 – Infrastructure

- Linux/git
- Conda-forge
- Jupyter/Jupytext
- Cocalc
- nbgrader
- nbsphinx
- pangeo

# Part 2 – Examples

- 2nd year course, 48 students
  - Jupyter/chrome/cocalc
- 3rd/4th year course, 26 students
  - Jupyter/laptops/conda-forge
- Grad/postdoc workshop, 20 students,
  - pangeo with dask/xarray,zarr on an HPC cluster

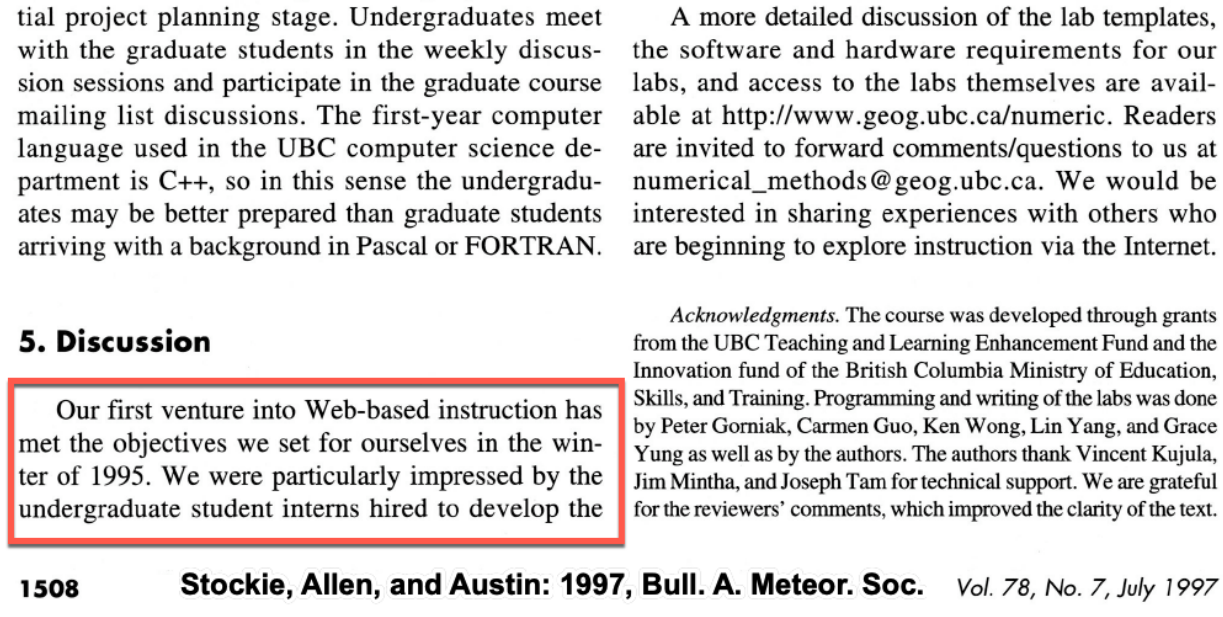
# Some history

**July, 1997: BAMS article: John Stockie, Susan Allen, Phil Austin**

*An Interactive Course in Numerical Methods for the Earth Sciences*

In [2]:
```
Image(filename='figures/bams_ref.png')
```

Out[2]:



tial project planning stage. Undergraduates meet with the graduate students in the weekly discussion sessions and participate in the graduate course mailing list discussions. The first-year computer language used in the UBC computer science department is C++, so in this sense the undergraduates may be better prepared than graduate students arriving with a background in Pascal or FORTRAN.

A more detailed discussion of the lab templates, the software and hardware requirements for our labs, and access to the labs themselves are available at http://www.geog.ubc.ca/numeric. Readers are invited to forward comments/questions to us at numerical_methods@geog.ubc.ca. We would be interested in sharing experiences with others who are beginning to explore instruction via the Internet.

## 5. Discussion

Our first venture into Web-based instruction has met the objectives we set for ourselves in the winter of 1995. We were particularly impressed by the undergraduate student interns hired to develop the

*Acknowledgments.* The course was developed through grants from the UBC Teaching and Learning Enhancement Fund and the Innovation fund of the British Columbia Ministry of Education, Skills, and Training. Programming and writing of the labs was done by Peter Gorniak, Carmen Guo, Ken Wong, Lin Yang, and Grace Yung as well as by the authors. The authors thank Vincent Kujula, Jim Mintha, and Joseph Tam for technical support. We are grateful for the reviewers' comments, which improved the clarity of the text.

# Infrastructure

## Linux kernel

- developed by 19,000 unique committers
- 65,000 files
- 25 million lines of code

## Git and Github

- 2005: Linus Torvalds (https://en.wikipedia.org/wiki/Linus_Torvalds) begins Git development (https://en.wikipedia.org/wiki/Git)
- 2008: Github founded (https://en.wikipedia.org/wiki/GitHub), in 2018 it hosted 28 million users and 57 million repositories
- Distributed version control makes possible the
  - pull request workflow (https://yangsu.github.io/pull-request-tutorial/)

## Conda-forge

- Software packaging using conda (https://conda.io/projects/conda/en/latest/index.html)
- Supported by numfocus (https://numfocus.org/project/conda-forge) -- provides scientifc software for linux, macos and windows 10 ~ 3500 separate packages
  - netcdf-fortran feedstock (https://github.com/conda-forge/netcdf-fortran-feedstock)
  - gdal feedstock (https://github.com/conda-forge/gdal-feedstock)

## Jupyter and jupytext

- Jupyter (https://blog.jupyter.org/jupyter-receives-the-acm-software-system-award-d43
  - Origins -- began in 2001 as ipython: repl (read, evaluate, print, loop)
  - Hosts 100 language kernels (https://github.com/jupyter/jupyter/wiki/Jupyter-kerne
- jupytext (https://github.com/mwouts/jupytext)
  - round trip conversion between jupyter notebooks written in javascript and
    - python, R, markdown, Julia, bash scripts
    - Allows collaborative notebook development uisng pull-requests
    - Example: this talk (https://github.com/phaustin/mcgill_2019_talk/blob/master/talk_dir/python/m

## Cocalc, nbgrader and nbsphinx

- cocalc.com (https://cocalc.com): juypter notebook hosting for classes
- nbgrader (https://github.com/jupyter/nbgrader): autograded jupyter notebooks
- nbsphinx (https://github.com/spatialaudio/nbsphinx): build a webstie from notebooks

# Examples

## Inspiration

- Cliburn Chan's Stat 663 (http://people.duke.edu/~ccc14/sta-663-2018/index.html)
- Berkeley's data 8 (https://data.berkeley.edu/education/courses/data-8)

## Large class, 2nd year

- Computational Methods in Geological Engineering (https://phaustin.github.io/eosc213/)
  - co-taught with Roger Beckie and Nicolas Seigneur
  - 48 2nd year geological engineering students
  - Prereqs: vector calculus, linear algebra, C
  - students use: cocalc.com notebooks

## Smaller class, 3rd and 4th year

- 3rd/4th year: Radiation and remote sensing (https://phaustin.github.io/a301_code/)
  - taught to 26 students in EOSC and other departments
  - open to any student with 1st year calculus, physics, programming
  - includes both jupyter notebooks and course libraries
  - students ues: conda environments on laptops or lab desktops

## Grad/post-doc workshop

- Parallel python tutorial (https://phaustin.github.io/parallel_python_course/)
- Introduce the pangeo software stack (https://docs.google.com/presentation/d/1evNXCddIIlXUt4a5jKfmlO2197sp8T6I9L650 particularly dask (https://medium.com/pangeo/dask-jobqueue-d7754e42ca53), xarray (h zarr (https://zarr.readthedocs.io/en/stable/). (see this pangeo notebook server (https://m binder-2ea923feb34f®))
- 20 UBC HPC focussed grad students/postdocs
- Half day, using notebooks and Compute Canada cluster

# Take home points

Jupyter notebooks plus conda-forge and github.

- Make active learning possible at scale. Costs to students are similar to the price they are paying for textbooks (cocalc plus a tablet/chromebook)
- In-class sessions and assignments deal with working examples of realistic problems and best-practices for production computing
- Use the same workflow for both research and teaching, and make research reproducible by default.

# What's next?

- I am writing an instructor's guide to teaching with Jupyter notebooks
- EOAS is creating a github repository for collaborative notebook development focussing on UCAR/ACCRU institutions, watch the UCAR members/ACCRU mailing lists
- Questions/interest/want to help? Email me at paustin@eoas.ubc.ca

*"The future is already here — it's just not very evenly distributed." -- William Gibson*