Computational Environments and Toolchains

Kieran Murphy and David Power

Department of Department of Science,
INSTITUTION.
(kmurphy@wit.ie,dpower@wit.ie)

Autumn Semester, 2021

RESOURCE OUTLINE LABEL

- Aim and motivation for this module
- Administration trivia — Contact hours, Assessment structure, . . .
- Tools and resources
- COVID arrangements

Aim of Module

1. Develop a working knowledge of:
   - networks and data communications.
   - operating systems and virtualisation solutions.
   - Linux terminal interface, such as for data retrieval, text file manipulation, and scripting.
   - modern programming languages (such as Python) in the retrieval and analysis of scientific data.
2. Build, configure and manage essential network infrastructure and application services.
3. Construct a pipeline of different programs that automates collation, analysis, and reporting of data.

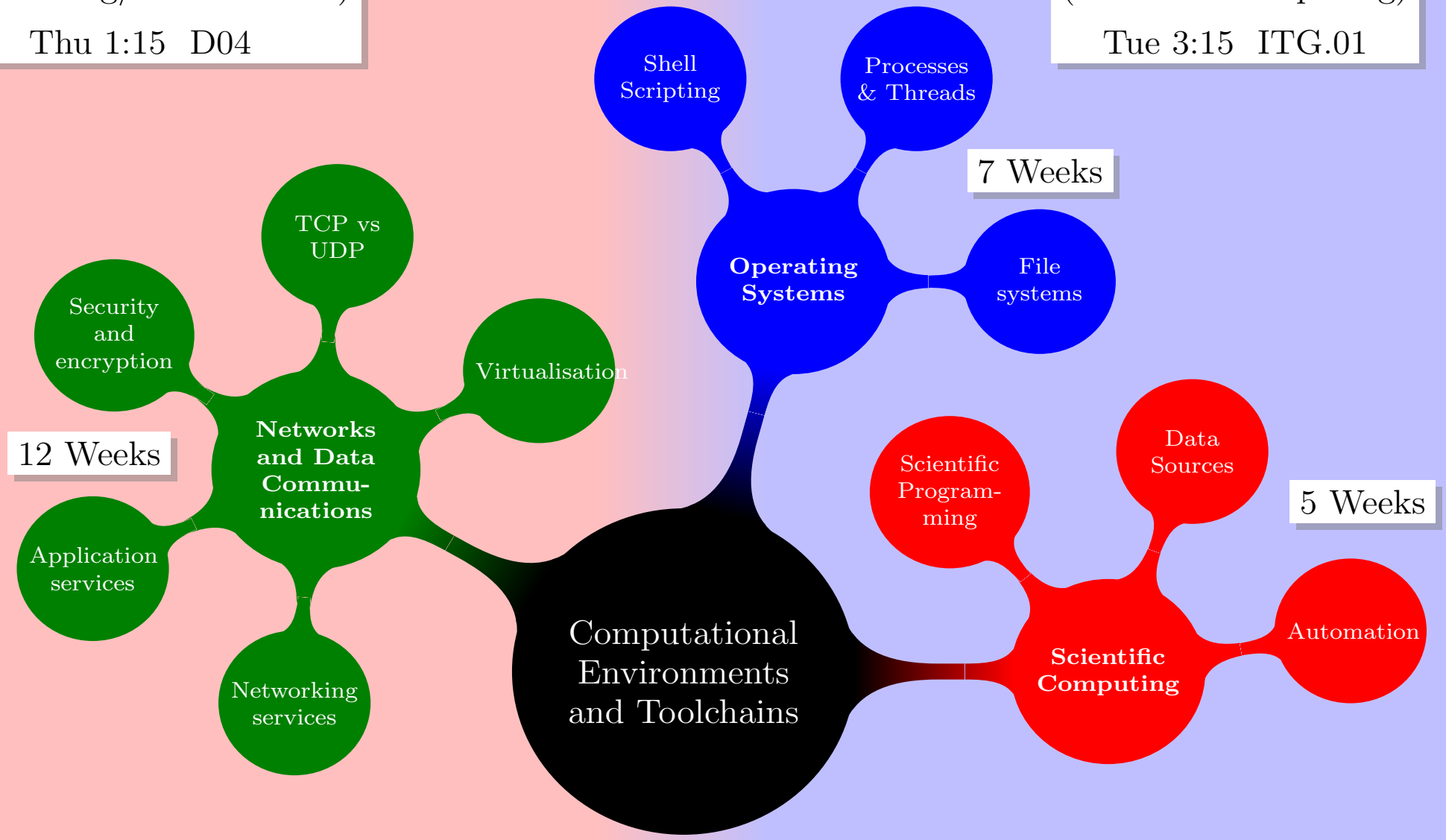
Informal Aim

Given a task;
1) determine required software to address problem,
2) **independently** investigate, build, configure and use selected software to solve problem, and
3) generate suitable reports

What? (and Who?)

David Power
(Networking/Data Comms)
Thu 1:15 D04

Kieran Murphy
(Scientific Computing)
Tue 3:15 ITG.01



Why?

This is a relatively new module* to address particular needs of Physics students in light of recent trends in the scientific computing world.

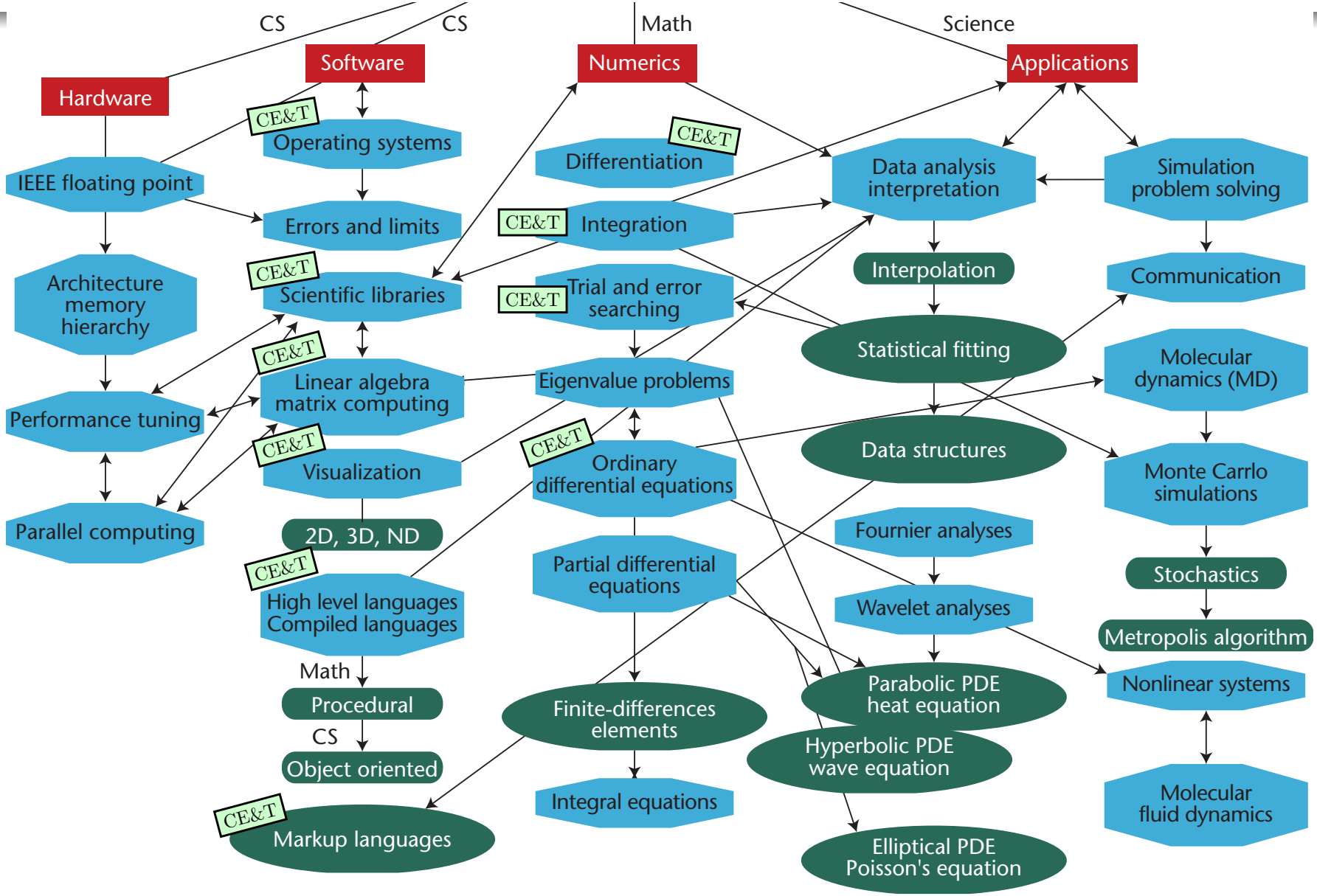
- Increasing use of virtualisation (and containerisation) to address networking, instantiation of compute nodes and services, and avoiding cross-platform porting of software.
- Industry, following the academic community, is moving from propriatorey to open source software for computational needs . . .
  - Computational Languages:
    MATLAB, Mathematica, Maple $\rightarrow$ Octave, **Python**, SCILAB
  - Statistics, Data analytics:
    SPSS, SAS $\rightarrow$ **Python**, R
  - Documentation generation, version control, etc
- Scientific community is moving towards reproducible research† — where both data and the code used in the analysis is also published.

*adopted from the existing Data Communications module shared with the BSc in Applied Computing students.

†10 Simple Rules for the Care and Feeding of Scientific Data

A Computational Physicist needs to Know Everything

. . .



How?

Contact hours

Two 2-hour lecture/practical sessions per week . . . . . (KM / DP)

David will talk to you about his approach for his lecture/practical sessions.

In terms of my sessions (Tue 3:15–5:00):

- Initially we will keep to same time slots, but I'm happy to move them around to suit the class as long as it does not clash with other commitments — *there has to be some advantage to being online.*
- Aim to structure lecture/practical sessions as:
  - Emphasis on practical work (read theory in own time).
  - Single/double practical for individual technology/topic/case study.

Learning Technologies

Moodle: moodle.wit.ie

- Launch point for module material.
- All assignments and module deliverables.
- Will copy-post from slack important message here.

Website: kmurphy.bitbucket.io/modules

- Location of module content.
- Links to deliverables.

Google's Colab: colab.research.google.com

- You can use your own instance of python (I prefer anaconda) and the Jupyter interface. However, to simplify things this semester I'm going to use Google's colab interface when working with python notebooks.

Slack: computational-yar6742.slack.com

- Used for instant messaging, one-on-one sessions, etc.

Assessment Structure

Module is 100% CA with 50% for Networking (DP) and 50% for Scientific Computing (KM).

David will talk to you about his approach for 50% for Networking.

In terms of my 50%:

Participation during in-class practicals, 20%

Taking online course requires more effort to actively manage one's participation and avoiding falling behind in the course work. In recognition of this I'm willing to allocate 20% of the module grade to deliverables that are intended to encourage participation rather than focusing on testing competency.

- Zoom/Slack polls, Moodle quizes (Engegement)

Computational Tasks, 80%

1. Moodle quizzes (Computational)
2. 4–5 Computational tasks (current plan)
3. Grade based on:
   - Level of specification that was satisfied
   - Analysis/programming quality

Academic Integrity

Official statement

The School of Science & Computing at Waterford Institute of Technology is committed to maintaining the highest standards of academic integrity. Academic misconduct, including but not limited to cheating may result in a mark of zero for the assignment as well as disciplinary action. Plagiarism or cheating may impact your grant if you receive one. Additional sanctions may by imposed depending on the case. You are responsible for adhering to all regulations regarding academic integrity. When in doubt about whether something is plagiarism, please see the guidelines published on the WIT website. ”

My comments or ‘lessons learnt from going online in Spring 2020’

- I have been involved in two cases of plagiarism in the summer exam sessions and it is not pleasant.
- Crossing the line from giving/receiving help to plagiarism is easy. So follow the rule of ‘writing every line of code yourself’ (even if you are getting significant guidance from other sources) and make sure you can explain every line.
- Most lectures will probably have an interview component to grading near end of semester.

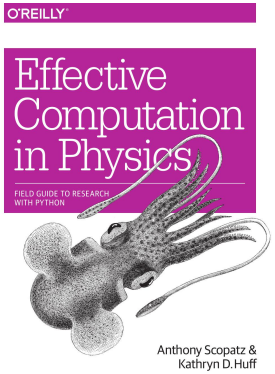
Netiquette

Remote learning can be taxing on the Student as well as the Lecturer. To assist with this, certain behaviours and standards are expected for online sessions with this module. When engaging with synchronous (e.g. live, real-time) remote classes

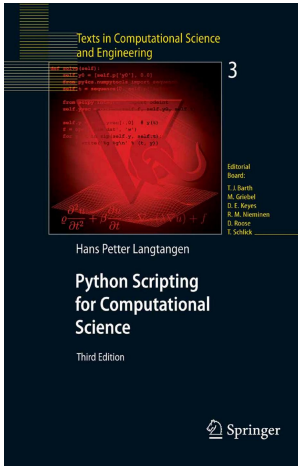
- Find quiet place free from distractions (siblings, pets, parents, television, etc).
- You will need writing materials for taking notes — you will not be able to record notes on a computer in real-time.
- Use headphones, mute your microphone and turn your camera on:
  - Video on encourages more active participation and promotes focus.
  - Zoom has a walkie-talkie feature — press and hold space bar while talking to be heard.
- Be respectful to your classmates and lecturers when speaking/writing.
- Attendance will be taken in every class.

Background Reading (OS & Scientific Computing)

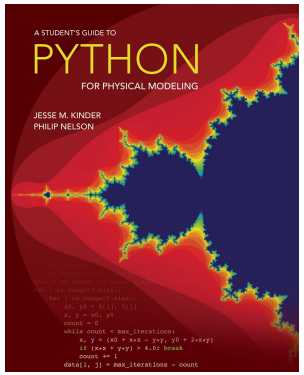
We will put up more texts later in the module, but just in case you are looking for some light reading to get you started have a look at these ...



Effective Computation in Physics
by Anthony Scopatz & Kathryn Huff
Touches on nearly all of the topics that we hope to cover in the Scientific Computing part of the course. Well worth buying.



Python Scripting for Computational Science
by Hans Petter Langtangen
Chapters 1 to 4 give a good coverage of the Python language, later chapters cover interfacing python with C/C++ (not needed).



Student's Python Guide
by Jesse M. Kinder & Philip Nelson
Good overview of python and we will probably base some of our practicals on those covered in this book.

Today's task — getting used to python/colab . . .