Computational Environments and Toolchains

Topic 00 : Module Overview and Introduction

Lecture 01 : Module Overview

Kieran Murphy and David Power

Computing and Mathematics, SETU (Waterford).
(kieran.murphy@setu.ie, david.power@setu.ie)

Autumn Semester, 2024/25

Outline

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

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.

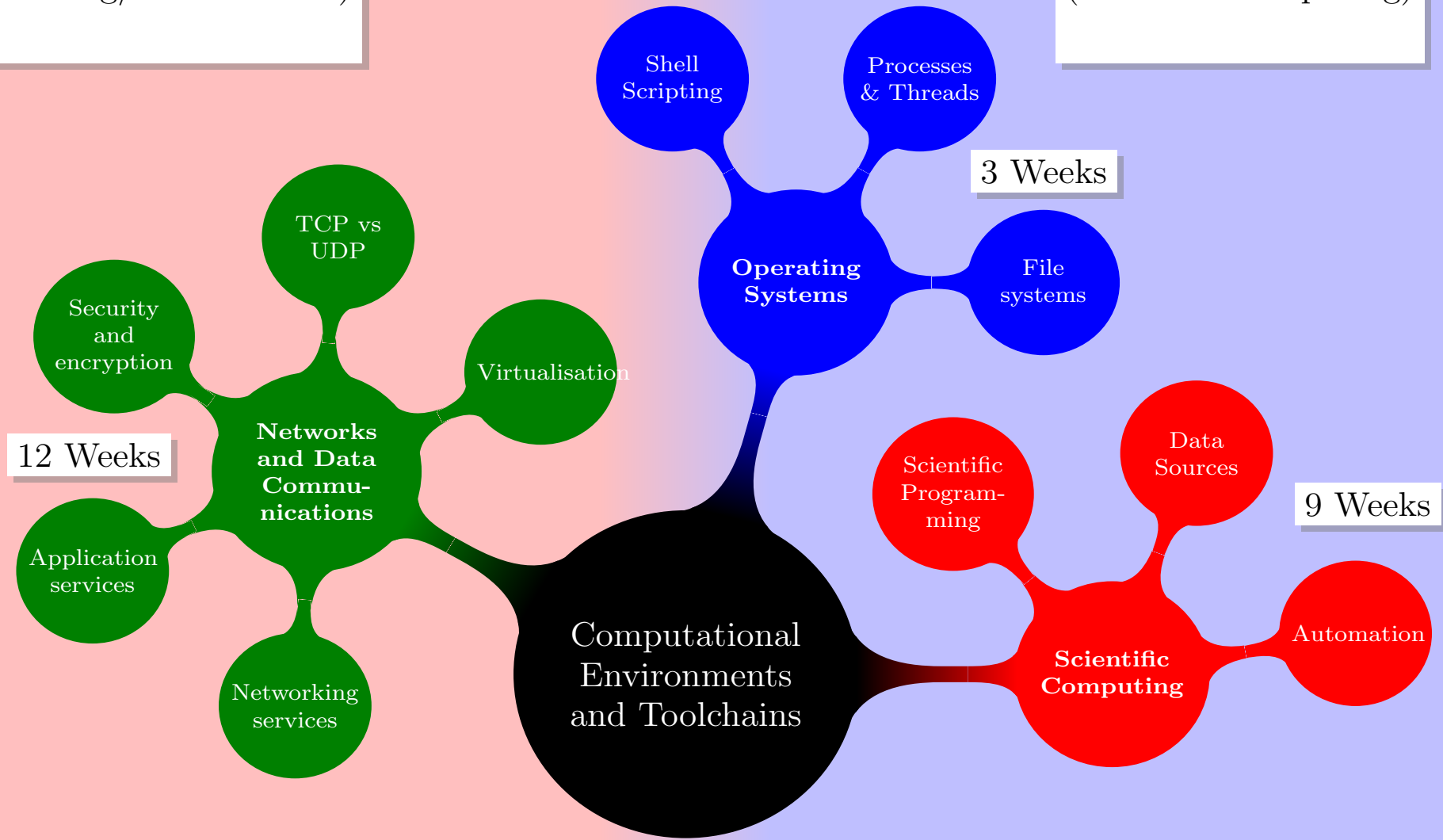
Informal Aim

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)

Kieran Murphy
(Scientific Computing)



Why?

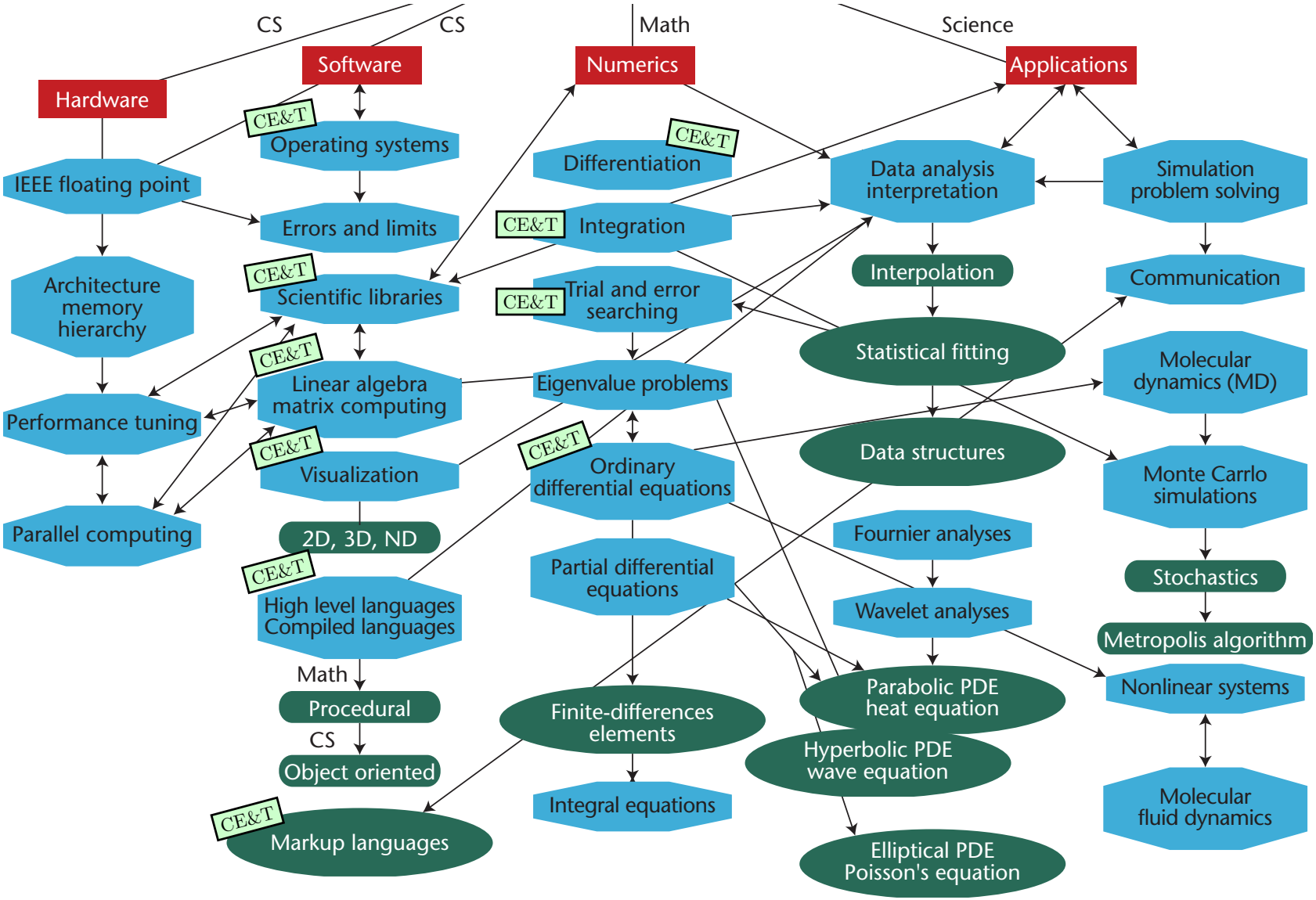
This is a relatively new module* to address particular needs of Physics students in light of recent trends in the scientific computing world.

- 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 → Octave, **Python**, SCILAB
  - Statistics, Data analytics:
    SPSS, SAS → **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 (arxiv.org/abs/1401.2134)

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 9:15–11:15):

- 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: setu-ceandt.github.io/live

- 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: ceandt202425.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%

- Student notebooks based on practical sessions.

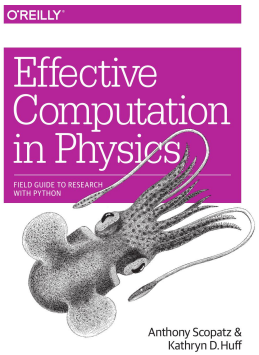
Computational Tasks, 80%

Weekly assignment activity (default due on Saturday @ 23:00):

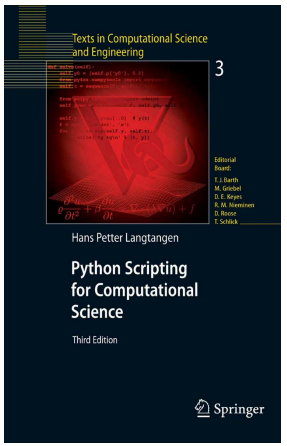
1. Moodle quizzes (Computational)
2. 4–5 Computational tasks (current plan)
3. Grade based on:
   - Level of specification that was satisfied
   - Analysis/programming quality

Background Reading (OS & Scientific Computing)

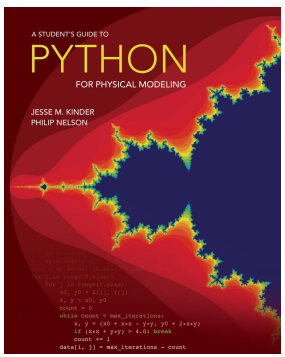
We will put up more texts later in the module, but just in case you are looking for some light reading to get you started have a look at these . . .



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.



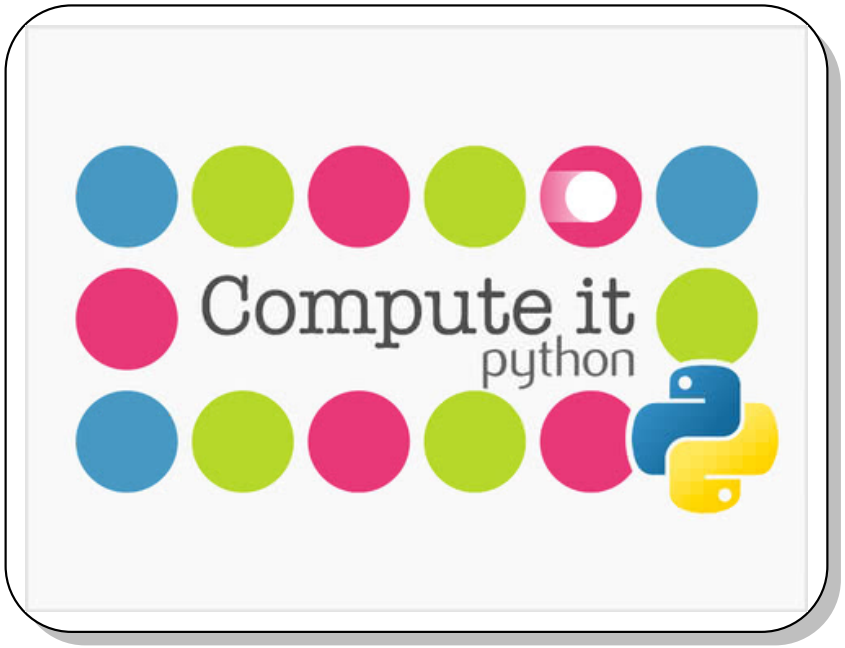
Python Scripting for Computational Science
by Hans Petter Langtangen
Chapters 1 to 4 give a good coverage of the Python language, later chapters cover interfacing python with C/C++ (not needed).



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 (W01) task — getting used to python/colab ...

Today's (W01) assignment — python hour of code activity



. . . and Moodle assignment link to upload progress . . .