Expanding the mental self-image of undergraduate Physicists

Ethics, coding, and context for the discipline

Sam Skipsey
for
APE 5th Birthday Event

What do I do?

P2T Level 2, 10 cr

"Coding" and Linux *for Physic*

75 students 50/50 CS/Phy

Assessment

1+6 Labs (in 3 sittings each
1 Practical Test (open book)
1 Written Exam (closed boo

I...

Write: Labs, Lectures
Run: (all) Labs
Give: ~25% of Lectures

Julia + Julia-HEP

4 workshops
Many papers

cross discipline

Supported by the HSF
Analysis and simulation tools
Integration with C++ codes

I...

Write: workshop material
Present: workshops
Run: benchmarking

CoP

olonising the Curriculum

) members

cross discipline
+ cross role

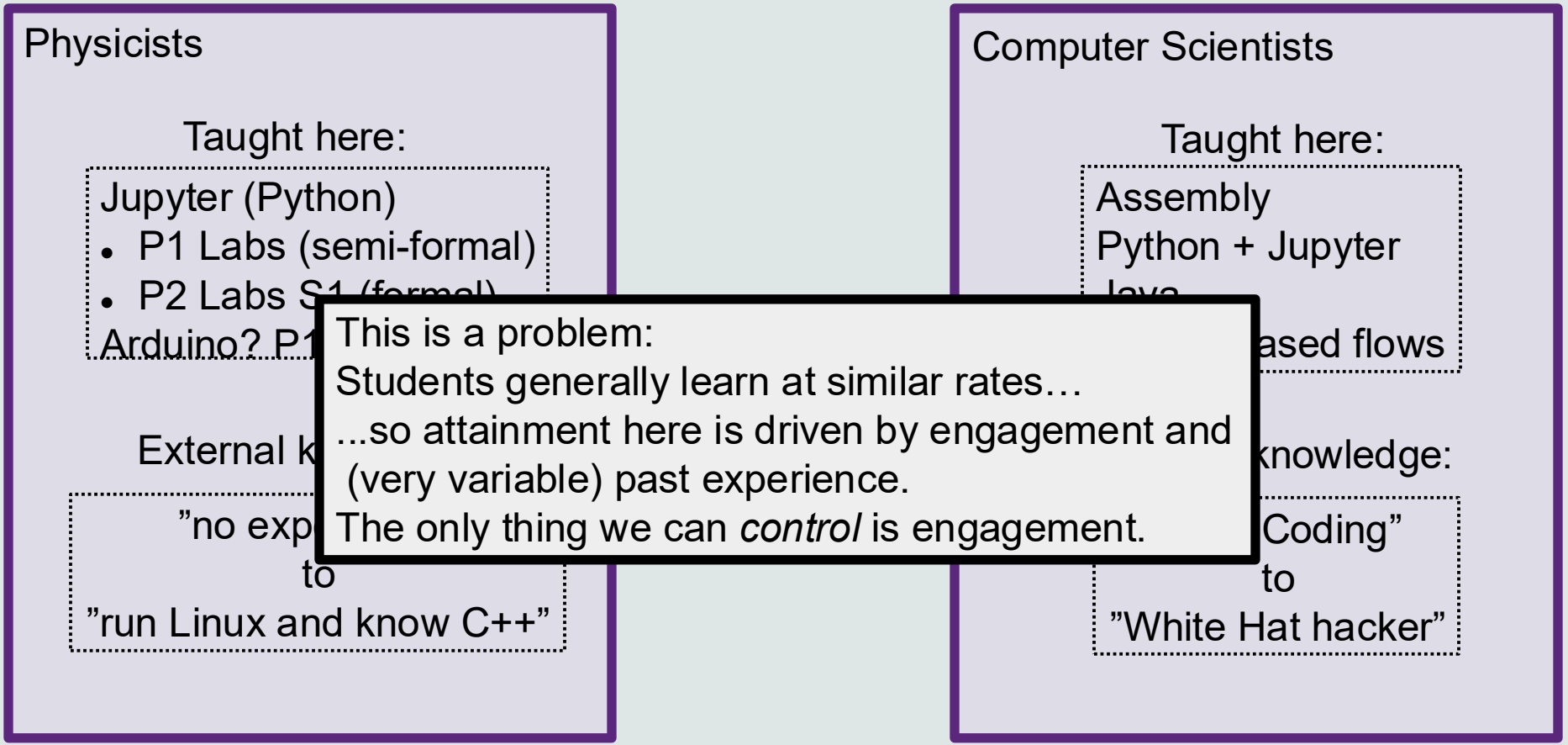
sources for Decolonisation
ordination
iversity Strategy!

I...

-lead: the CoP
-run: student projects,
ding group
angelise!

P2T: (C/Python) Programming Under Linux

P2T: Diverse programming backgrounds



P2T: Justification

“Why should I, a Physicist, care about coding?”

Physics, as a discipline, rarely justifies why students should care about particular modules. (The justification is inherent: it’s important *because* we’re teaching you it.)

This does not work for coding/programming – students seem to have mental models of “physicist” which do not include programming skills as a core component.

(but: note that not all students doing a degree in Physics see themselves as Physicists...

…and government messaging about the "point" of degrees often doesn't help here.)

P2T: Justification

“Why should I, a Physicist, care about coding?”

Why *C (and programming)*?

- Isn't Physics just maths and experiment?
  - "just maths" covers a multitude of sins.
  - ("just experiments" also covers a lot of things.)
- In general, "numerical" work - which used to be done by *human* computers - now is done via computer programs.
-

Zeros of polynomials

Experimental Data

- Analysis
  - "Data Science"
  - This is the more complex version of what you've been using numpy + python for in P1, P2 Labs
  - Machine Learning (“AI”) is also increasingly important in many areas of Physics.

22ND IEEE REAL TIME CONFERENCE, 12-24 OCTOBER 2020 1

New software based readout driver for the ATLAS experiment

Serguei Kolos, on behalf of the ATLAS TDAQ Collaboration

Machine Code (~400BCE to 1947CE)

For thousands of years, people have made mechanical devices to perform tasks or calculations. All of these were "programmed" by directly manipulating cogs, gears, pins, etc to change the pattern of how they worked; or by patterns of holes in cards which would move pins as they passed over them.
This "low-level" programming is very hard to do without mistakes.

Assembly Languages (1940s-)

Left: Betty Jennings and Frances Bilas pictured (posed for the photograph) programming the ENIAC (~1947)

```
// /CT/
E/ DSTI
@/ D//H
A/ R//P
:/ /C/S
S/ :CT/
I/ @CTI
U/ :C/S
¼/ JS/P
D/ A/
R/ @/
```

Right: The example program provided by Alan Turing for Manchester's Mark II computer (c1951).
(It multiplies two numbers!)

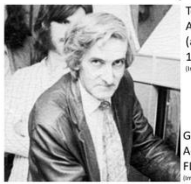
Below: The same program, but as a generic assembly language.

```
      MOV 0, ACC
LABEL1:CMP 0, R2
      JZE LABEL2
      ADD R1, ACC
      SUB 1, R2
      JMP LABEL1
LABEL2:STP
```

Kathleen Booth invents concept of Assembly language (~1947-1949)
Allows humans to write "easily memorable codes" (eg ADD for an instruction that causes addition), and have them replaced by the correct set of holes in a tape (eg).

This is still finicky, but at least it's not as error prone as direct machine-code programming - it is much easier to remember "ADD" than a sequence of holes in a punched card....

Compiling (and Linking) (1950s-)



Tony Brooker - Autocode (1954) (after Alick Glennie 1952)
(Image of unknown status)

Grace Murray Hopper A-0 (1951), FLOW-MATIC (1955)
(Image: CC BY 2.0 Jan Arkesteijn)

John Backus - FORTRAN-1 (1954)
(Image of unknown status)

The concept of a "high-level" language, which could be automatically converted into more numerous "assembly" instructions, or directly to machine code develops in the 1950s.
A "compiler" does this conversion - originally, working by directly replacing "commands" with pre-set lists of assembly instructions.

At the same time, people also wanted to be able to re-use other people's code (rather than rewriting some code to do the same thing each time).
"Linkers" were developed to allow pre-written solutions - stored in literal libraries of code on punched tape - to be automatically added to a program that needed them.

```
let x = 0
given integers a,b
while a is not 0:
 add b to x,
 subtract 1 from a.
stop.
```

P2T: Confidence, Plagiarism & LLMs

- I've studied successive cohorts in terms of:
  - Where they tend to "cheat"
  - *How* they tend to "cheat"
  - Correlations between "cheating" and performance
- Lack of coding as part of the "physics" mental image also contributes to plagiarism and academic misconduct.
  - Students don't cheat on things that they *want* to get better at
  - Engagement is the best solution to academic misconduct
- But "perceived difficulty" can drive other kinds of "cheating"
  - Proactive and engaged demonstrators are critical to supporting students starting from a lower level of experience!

P2T: Confidence, Plagiarism & LLMs

- How do students “cheat”?
  - “Performance enhancing drugs” (Nootropics)
  - “Homework help sites”
  - LLMs

Justifications are the same in all cases:

“there isn’t enough time to meet the requirements without help”.

A Sense of *Unfairness*.

“Everyone else is doing it” [a Red Queen’s Race]

particularly for LLMs, driven by pervasive advertising from the *companies that push them* (and Government pushes, due to lobbying from those same companies)

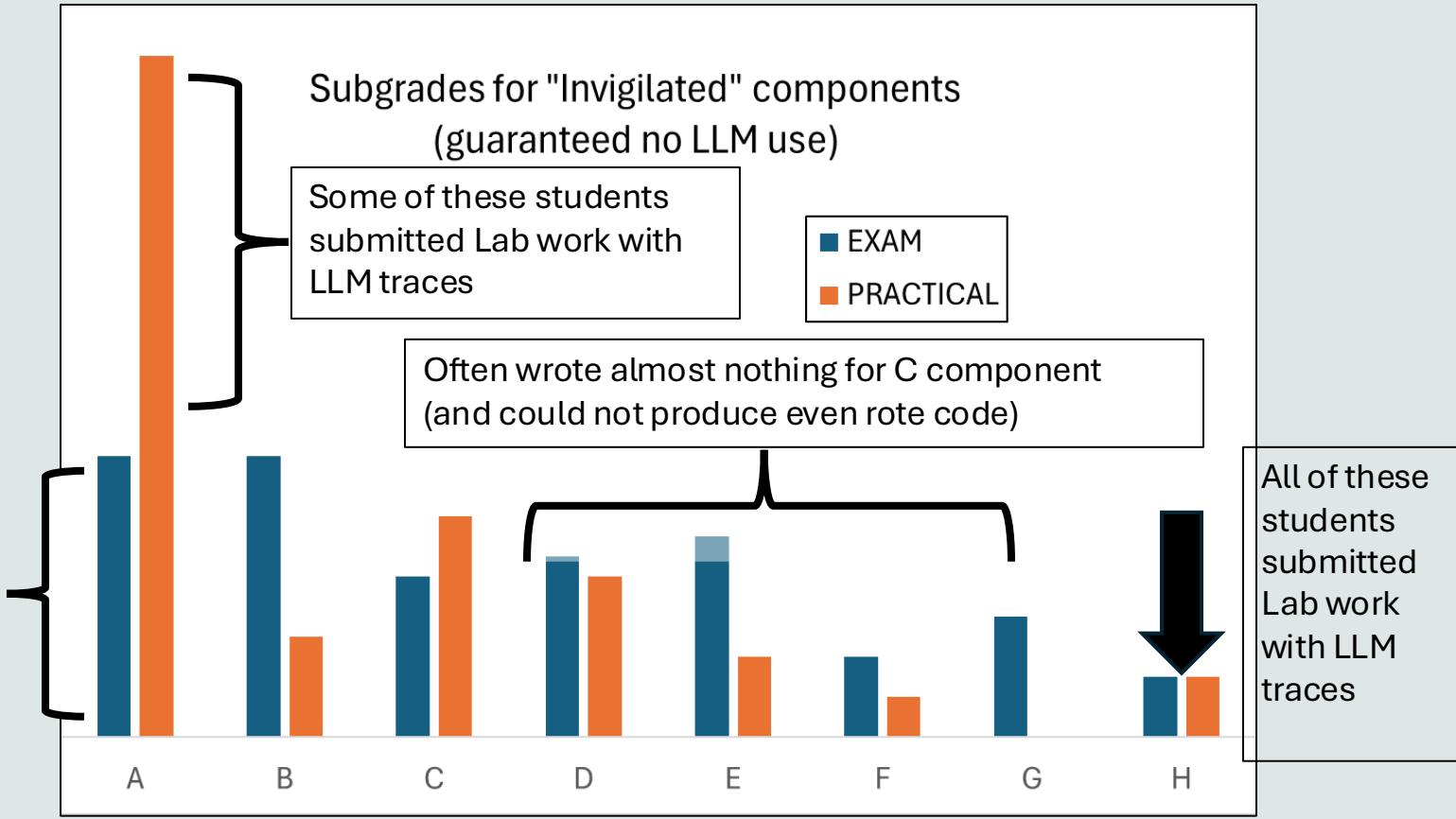
P2T: Confidence, Plagiarism & LLMs

- Two kinds of “cheating”:
  - Disengagement from the material (“I just need the grade, someone else can do the work”)
  - Performance enhancement (“I want to be able to do this, but I need assistance to be better”)
- LLM “cheating” looks like homework help cheating – *disengagement* from the material
  - As such, it can be made much easier to detect by designing assignments so that the “headline” for the assignment produces a different result to following the details of the task.
  - (Ironically, this also helps us scaffold the task for those actually engaging with it…)
- We also see LLMs used as “cognitive crutches” in Labs
  - This is more similar to nootropic use, but with the

P2T: Confidence, Plagiarism & LLMs

LLM use reduces *engagement* – in common with automation in general.

Reduced engagement = reduced learning / internalisation.



Programming Pedagogy (Julia)

P2T: Programming Pedagogy

The best programming language is (sub)*discipline*-dependent.

Expressing Physics ideas in Python is *unnatural* – we need numpy (etc) to actually "talk Physics".

Some experiments in teaching *Julia* as an alternative language…

In general, ~~guinea-pigs~~ students with Math/Physics backgrounds find Julia more natural to learn, due to syntactical choices.

P2T: Programming Pedagogy

Breaking down boundaries:

- Summer 2025 workshop @ UoG
  - Invited Staff and Undergraduate cohort equally.
  - Both cohorts treated identically in the workshop
    - More discussion oriented than most

Post event Feedback from undergrad cohort was positive –

- encouraged to do future work in Physics dept

- enthused by Julia as a language.

Decolonising (the Curriculum)

Decolonising The Curriculum

"The effort to interrogate and transform the institutional, structural and epistemological legacies of colonialism"

[The Decolonising SOAS Working Group]

Decolonising

This is an institutional level problem directly relevant to learning, teaching, *and research* through

- **Developing critical thinking and challenging knowledge**
- **Tackling a diversity of real-world problems incorporating global perspectives**
- **Providing a richer and nuanced understanding of the world**
- **Making space for equitable institutional processes and structures**

Decolonising: Curriculum For Life

C4L courses are credit-bearing, elective modules designed to help students engage meaningfully with learning, knowledge, and ways of thinking and practicing beyond their core disciplines.

Aligned with the UN Sustainable Development Goals (SDGs), these courses focus on real-world challenges, foster future-ready skills, and encourage collaboration across subjects, Schools, and degree programmes, providing students with opportunities to develop themselves as global citizens, creative leaders, change makers and social innovators.

Decolonising: Curriculum For Life

“Science in Context”*

- Co-proposed by: Sam Skipsey, Colette Mair (Statistics), Henry Ivry (Critical Studies), [Sarah Cockram (History)]**
- With feedback from: Michael Hicks (Philosophy), John Donaldson (Philosophy)

Introducing students to both how STEM fields can influence (and have been cited to influence) culture, politics, etc ; and also how culture, politics etc influences what STEM research is possible or conceivable.

Accepted for development (after this talk was given)

*Working title – we need to find something that doesn’t scare CoSE students off	**Sarah is one of the original co-proposers, but can’t allocate time to it this AY. We’re keeping her in the loop, with her permission.

Summary

Broadening the “box” of “Physicist” and “Physics curriculum” is a net gain for many students

Physics (and STEM) must include *ethics*, and “subjective” aspects to be part of society.

Physics courses must also reflect everyone in society to engage students fully.

LLM use is both a sign of disengagement, but also pernicious due to huge promotion from vested lobbying interests… and any putative "positive" use must reckon with the easy availability of the negative uses regardless.

References

Programming Pedagogy, the effect of Automation on Engagement, Students reasons for "Cheating" by enhancement

https://arxiv.org/abs/2506.08872 "Your Brain in ChatGPT: Accumulation of Cognitive Debt when Using an AI Assistant for Essay Writing Task" (2025)
https://www.microsoft.com/en-us/research/publication/the-impact-of-generative-ai-on-critical-thinking-self-reported-reductions-in-cognitive-effort-and-confidence-effects-from-a-survey-of-knowledge-workers/ "The Impact of Generative AI on Critical Thinking" (2025)
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4895486 "Generative AI can Harm Learning" (2024)
https://www.psychologytoday.com/us/blog/our-new-discontents/202309/what-to-do-about-student-cheating "The Real Roots of Student Cheating" (2023)
https://doi.org/10.1080/09687637.2018.1555231 "Developing expertise, customising sleep, enhancing study practices: exploring the legitimisation of modafinil use within the accounts of UK undergraduate students" (2019)
https://journals.sagepub.com/doi/10.1177/0956797613504438 "Point-and-Shoot Memories: The Influence of Taking Photos on Memory for a Museum Tour" (2013)
https://www.sciencedirect.com/science/article/pii/0005109883900468 "Ironies of Automation" (1983)

https://doi.org/10.1073/pnas.2221311120 "An astonishing regularity in student learning rate"
Nuthall, G. (2007) The hidden lives of learners. Wellington: NZCER Press

Julia

https://github.com/JuliaHEP/Hands-on-Julia-for-particle-physicists A Julia workshop for Particle Physicists
https://github.com/michiganrobotics/rob101 The Linear Algebra for Robotics course at U Mich, which uses Julia to teach linalg as a transparent language for expressing these things (compared to Python, say)

Decolonisation, Ethics in STEM

MSOR Connections 19(2) pp75-86 'Welcoming students to the Mathematics community: obstacles to "belonging"'
https://www.iop.org/about/IOP-diversity-inclusion/physics-inclusion-award
J. Chem. Educ. 2022, 99, 5−9 "Decolonizing the Undergraduate Chemistry Curriculum: An Account of How to Start"
https://www.gla.ac.uk/myglasgow/learningandteaching/staff-curriculumforlife/