ACS: Applying Computational Science

Gareth Collins /
James Percival

Group projects (ACS Applying Computational Science)

"Applying Computational Science and Engineering will give you a real-world experience of high-productivity problem solving."

Objectives

- To simulate applied computational science in the real world.
- To synthesise knowledge from the taught modules
- To apply techniques learned in the course to real problems
- To develop collaborative programming skills
- To reinforce best practise for software development

Project schedule:
Project 1: 20–24 November 2023
Project 2: 29 Jan–2 Feb 2024
Project 3: 20–24 May 2024

Assessment:

Software	(70%)
Presentation	(20%)
Teamwork	(10%)

Group projects (ACS Applying Computational Science)

Module Learning Outcomes:

- Plan and produce software collaboratively.
- Collaboratively solve problems using software.
- Summarise work using collaborative presentations.

A welcoming environment

- We encourage and promote diversity in science
- Whoever you are, and whatever your background, we welcome you.
- We hope that everyone finds their experience welcoming, encouraging and rewarding.
- We want to foster a community based on mutual respect, tolerance, and encouragement and we kindly ask that you respect these principles.
- If you experience or witness any unwelcome behaviour we encourage you to challenge the behaviour or report it, in confidence, to teaching staff, the module coordinator or the course director.
- To allow everyone to fully contribute, please use English in code comments and when meeting as a team (e.g. as 3 or more people).
- If you have questions about support or pastoral concerns, talk to PGT Senior Tutor: James Percival (j.percival@imperial.ac.uk)

IMPERIAL

DEPARTMENT OF
EARTH SCIENCE AND ENGINEERING

ACS Project Timetable

	Monday	Tuesday	Wednesday	Thursday	Friday
09:00	Lecture / Group assignment / Q & A	Group working	Group working	Group working	Group working
12:00					12:00 noon Code Deadline
14:00	Group working	Group working	Reserved For Sports	Group working	Group working
17:00					4:00 pm Pres. Deadline 4:30 pm Debrief

Software submission deadline: **Friday 12 noon;** Video presentation deadline: **Friday 4:00 pm**

Our expectation is that you will each spend approximately **45 hours** on these projects, over the course of the week (9 hours / day): The project should be your focus for the whole week.

ACS Project Support

- This project is about collaboration: your first port of call for help should be each other!
- We have created a **private channel on Teams** for each group to ask us questions
- We will hold a general Q&A session Tue/Wed/Thu mornings in 1.47 at 9.15 am
  - (send 1-2 representatives from your group with questions)
- We will schedule 10 min. meetings for each group for in-person meetings or video calls Tue/Wed/Thu to answer questions
- At other times we will try to answer questions via chat as soon as possible
- Please ask questions of **clarification** in the **General** channel
- This time there **will not be indicative scores** of performance during week

A word about academic integrity

- **Academic integrity** is fundamental to learning, teaching and research
- **Academic misconduct** is the attempt to gain an academic advantage, whether intentionally or unintentionally, in any piece of assessment submitted to the College
- Each project will provide specific guidance about which aspects of the software must be written from scratch and which aspects can make use of packages or code written by others.
- **Plagiarism must be avoided**. If you use code sourced outside your group, you must include clear and proper attribution of credit.
- In group projects, **collusion constitutes the sharing of work between groups**. You are therefore strongly discouraged from discussing the group project with peers outside your project group.
- For group projects, **you are actively encouraged to work collaboratively as a team**—sharing ideas and code within your team is exactly what you should be doing and is **not collusion**.
- **As a guiding principle**: always acknowledge the contributions of others in your work, and do not leave yourself open to allegations that you have supplied answers to enable another student or group to commit academic misconduct.

Use of Generative AI Tools

- Use of AI tools (e.g. ChatGPT, Copilot) is allowed, as long as you use them **responsibly**
- Remember not to trust everything generated by AI, and **test/verify** everything that they produce
- Any time that you use AI tools, you must add an acknowledgement/reference
- If you have used some code generated by AI in your project, you must add a note to the docstring for the relevant function to acknowledge the source of the code
- In your documentation, you should add a paragraph describing your group's use of AI tools throughout the project

Project 2 – The Day After Tomorrow – Feedback

- Overall: very successful – well done!
- Marks were good (average 68%; group marks in range 63-74%), but room for improvement!
- Things you did well:
  - Software functionality – notebooks and models were generally of a high standard and targeted well at the problem
  - Software sustainability – Utility code was generally well packaged, and thought had gone into the way your repositories worked
  - Group working (mostly!)
- Things to improve:
  - Presentations – **Leave time to rehearse** & make sure you fit the time slot. Start making presentation materials as soon as you can. Target your audience, you're explaining to your potential client (but do be honest). Explain your choices and why you made them. Think about data visualization
  - Everyone can and should make a substantial contribution: If you don't think your group is using you well, don't be afraid to say so in meetings. If people don't know what you did, they may not correctly assess your individual contribution.

ACS Project 3

The Gerardium Rush

RSM 1.51

- Acanthite
- Bauxite
- Cuprite
- Dolomite
- Enargite
- Fluorite

RSM 1.49/50

- Galena
- Hematite
- Ilmenite
- Magnetite
- Pentlandite
- Sphalerite

Rooms available all week

- 1.51
- 1.49/50
- 1.47 (morning briefings)

Available Thursday (all day) & Friday (am)

- 3.01 D/E

Presentation rooms/times for Friday pm

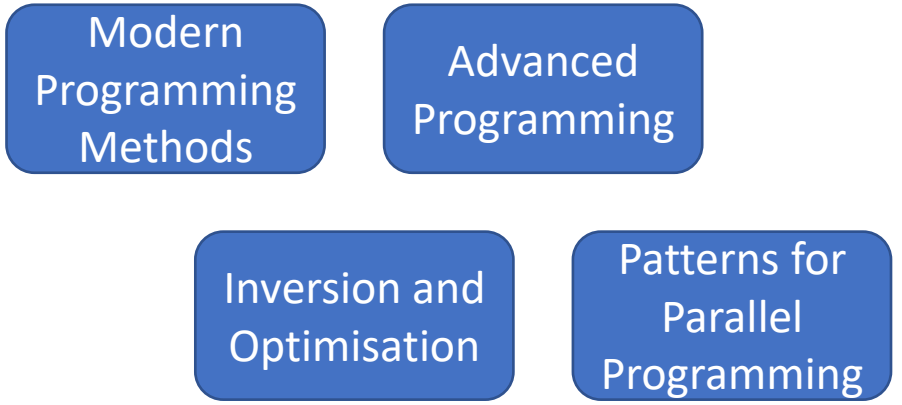
- Will be announced later in the week

Learning Outcomes

At the end of this exercise, you should have learned:

- To develop software collaboratively
  - The value of **automated testing**
  - Effective **teamwork**
- New technical skills:
  - Solve a **global optimisation problem**
  - Applying a **meta-heuristic** optimisation algorithm
  - **Interpret** model results

- Synthesising knowledge from:



- Programming with C++ & Python
- Using GitHub for collaborative software maintenance
- GitHub Actions for automated testing
- OpenMP/MPI for parallel processing
- Solve an optimisation problem

Final thoughts…

- This module is about collaboration: talk to each other; be kind to each other
- Choose a leader
- Make a sensible division of labour
  - Split into 3-4 sub-groups to solve each part of the problem
- Work in pairs (effective way to program and more fun!)

Questions we will answer:

- Clarifications about problem / requirements / assessment
- Help with Python / git / GitHub / GitHub Actions / Sphinx / C++

Questions we won't answer:

- Choice of algorithm / approach
- Help with debugging your code