# PX915-A Work Plan

Dylan Morgan, Matyas Parrag, Anas Siddiqui, Ben Gosling, and Geraldine Anis

April 27, 2022

## Team Organisation

The development of the code and documentation has been split into several sections given below:

- Make/Compile scripts – (Assigned: Dylan Morgan)
- User Input – (Assigned: Matyas Parrag)
- Backend/Algorithmic implementation – (Assigned: Anas Siddiqui and Geraldine Anis)
- Outputting data and Visualisation – (Assigned: Ben Gosling)
- Checkpointing – (Assigned: Matyas Parrag)
- Testing – (Assigned: Interim test - All members; Final Test - Anas Siddiqui and Ben Gosling)
- Parallelisation – (Assigned: Anas Siddiqui, Dylan Morgan and Geraldine Anis)
- Uncertainty Quantification (UQ) and Validation – (Assigned: Ben Gosling and Matyas Parrag)
- GitHub Maintainer – (Assigned: Dylan Morgan)
- Documentation Management – (Assigned: Geraldine Anis)

(Note: Every group member will document and write tests for the code that they have written. However, one person is designated to format and manage it).

Each group member has been assigned to sections of the code and document developments (as shown above), with Git issues given to members associated with their assigned tasks.

## Task Breakdown and Order

The task breakdown throughout the assigned time period is represented as a Gantt chart, displayed in Figure 1: where the chart is colour-coded such that:

- Anas – Blue,
- Ben – Red,
- Dylan – Violet,

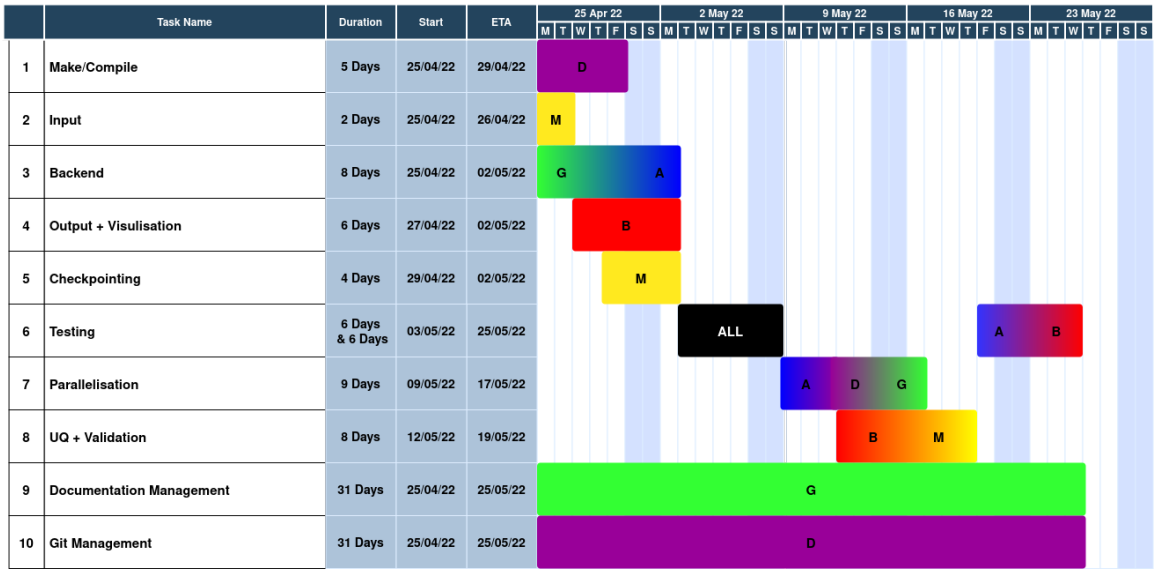


Figure 1: Gannt chart displaying the ordering and breakdown of the tasks to be completed within the given time frame. The colour code matches that defined in the main text, and the group members' first initial is labelled within the bars.

- Geraldine – Green,
- Matyas – Yellow .

Note: when any group member is idle or finishes a task ahead of time, they will help prepare for the following tasks or monitor other members' progress.

# Risks and Management

- **Risk:** Group member requires time off (due to illness, etc.)
  - **Mitigation:** The schedule will be reorganised to ensure the missing members' tasks are completed. This will be achieved either by splitting a job that is expected to be done by multiple people into being completed by an individual, or to reassign members that will be working on later tasks to cover the work of the absent individual.
- **Risk:** Parallelism isn't fully completed on time
  - **Mitigation:** Parallelism will be cut short to devote time and resources to developing a well-functioning serial code to ensure that we are able to have a working code to obtain data from.
- **Risk:** Code is not fully developed on schedule
  - **Mitigation:** Devote time and resources from UQ and validation to other areas to ensure the code is built on time. UQ and validation can be completed after the code delivery date, and for the date of the presentation.