

How do Novices Spend Time Programming in MATLAB?

Warren Code

Abstract

Learning to employ computers for data analysis is now an essential part of many degree programs, and for engineering students this can mean development of a working knowledge of MATLAB. At UBC as elsewhere, a common early learning environment for programming is a *computer lab*: time allocated to use provided machines, a list of activities or deliverables, and in-person TA support. Current literature suggests that certain patterns of work denote sophistication in novice programming ability. Study of this highly unstructured time can be a challenge, however, so here we present the development and trial results of an automated recording scheme that permits detailed tracking of student progress with essentially zero interference in student work flow.

Setting

- 2nd Year Mechanical Engineering students have two seven-week blocks (Oct-Dec and Feb-Apr) with weekly computer labs.
- About three-quarters of students have taken Math 152 (see other poster).
- Activities include working with data, visualization of data, plotting in 2D and 3D, and basic numerical methods for differential equations and integration.

Computer Lab Setup

- Students are assigned to one of Monday, Tuesday, Wednesday or Friday afternoons.
- On their lab day, students have two hours allocated in the lab with a TA.
- Students must submit their work by the end of the two hours.
- Collaboration is permitted, as long as the appropriate credit is given.

MATLAB

- MATLAB = Matrix Laboratory; a “high-level language and interactive environment” (according to the MathWorks website).
- Programming: direct interaction with the Matlab command line, or via “script” (list of commands) or “function” (method) files.
- Built-in methods for reading in data, working with vectors/matrices, plotting in 2D or 3D.

Previous Study

- In-lab, one or two students at a time:
 - observe students working
 - help with MATLAB issues to probe understanding
- Keep an approximate record of completion time and difficulties encountered (common errors and issues with activities).
- Interview TA about same.

Purpose of New Tool

- Determine carefully how widespread observed errors are among the whole cohort.
- Far more granularity of information about time spent in the lab and on each activity.
- Opportunity to “observe” work of the more capable programmers who would not ask for help in-lab; can compare to expected styles based on novice programming literature.

Capturing Work in MATLAB

- Built-in “diary” function captures sequence of commands and their outputs.
 - No timestamps.
 - Large text file may be generated by loop output.
 - No summary or analysis tool is provided.
 - Manual start-up and management of the generated diary files.

New Session Logging Program

- All MATLAB commands, runs in background (without user attention) to produce a log file.
- Uses MATLAB's data-acquisition "timer" object to sample every 30 seconds:
 - New commands and output in the diary file.
 - Changes in script and function files (.m extension) in the working directory.
- Saves to log file with timestamps.
- Non-invasive: other computer activity is not tracked. Current version does not record file contents, only times when the file was updated.

Implementation

1. Trial with built-in diary only (no timestamps, many files too large to transmit/process): one lab week in first term.
2. Trial with more complete logging system for two lab weeks in second term.

Trial 1:

Built-in Matlab diary

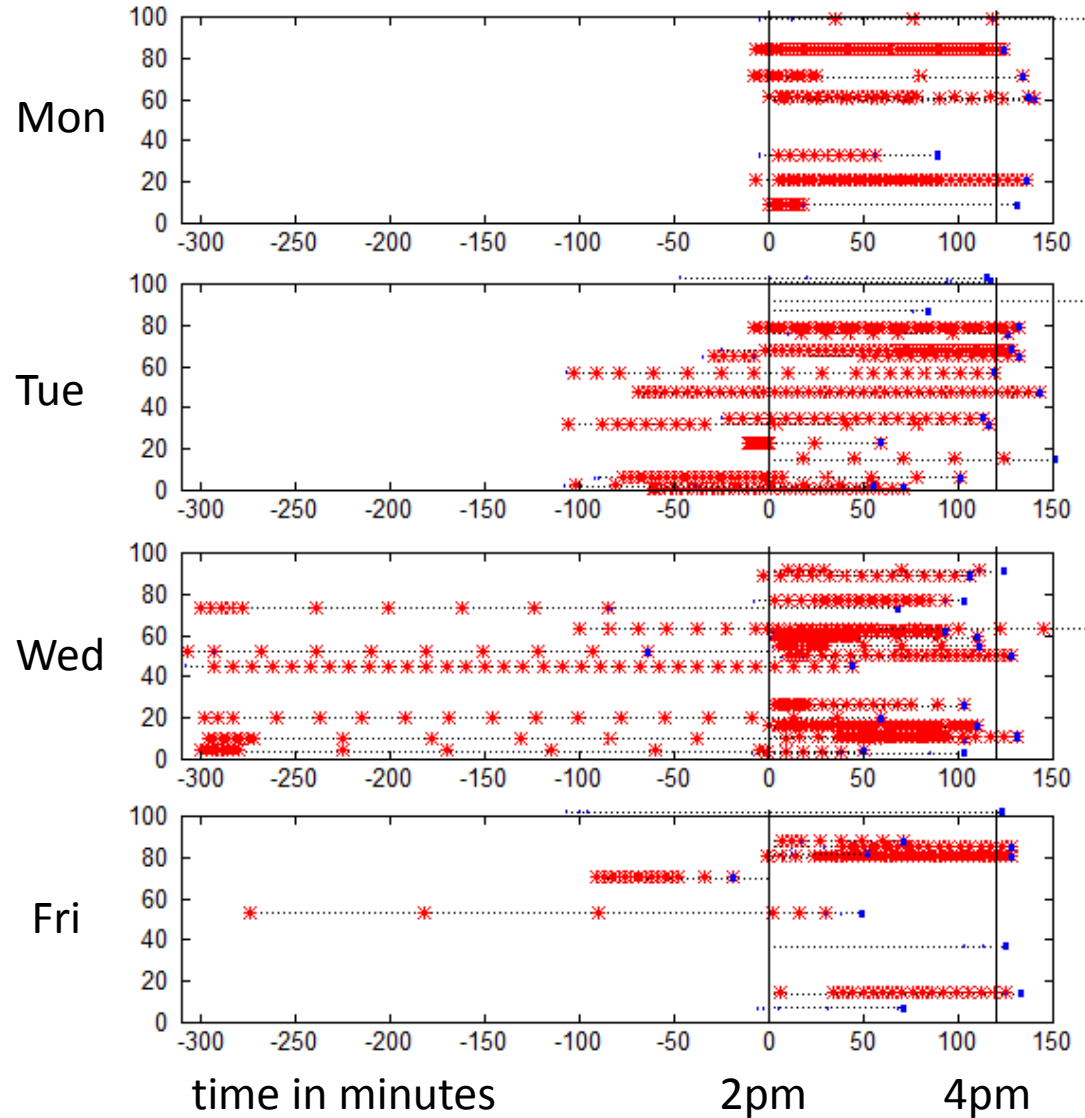
Assigned lab times are between 2pm and 4pm. Red asterisks indicate activity.

Learned:

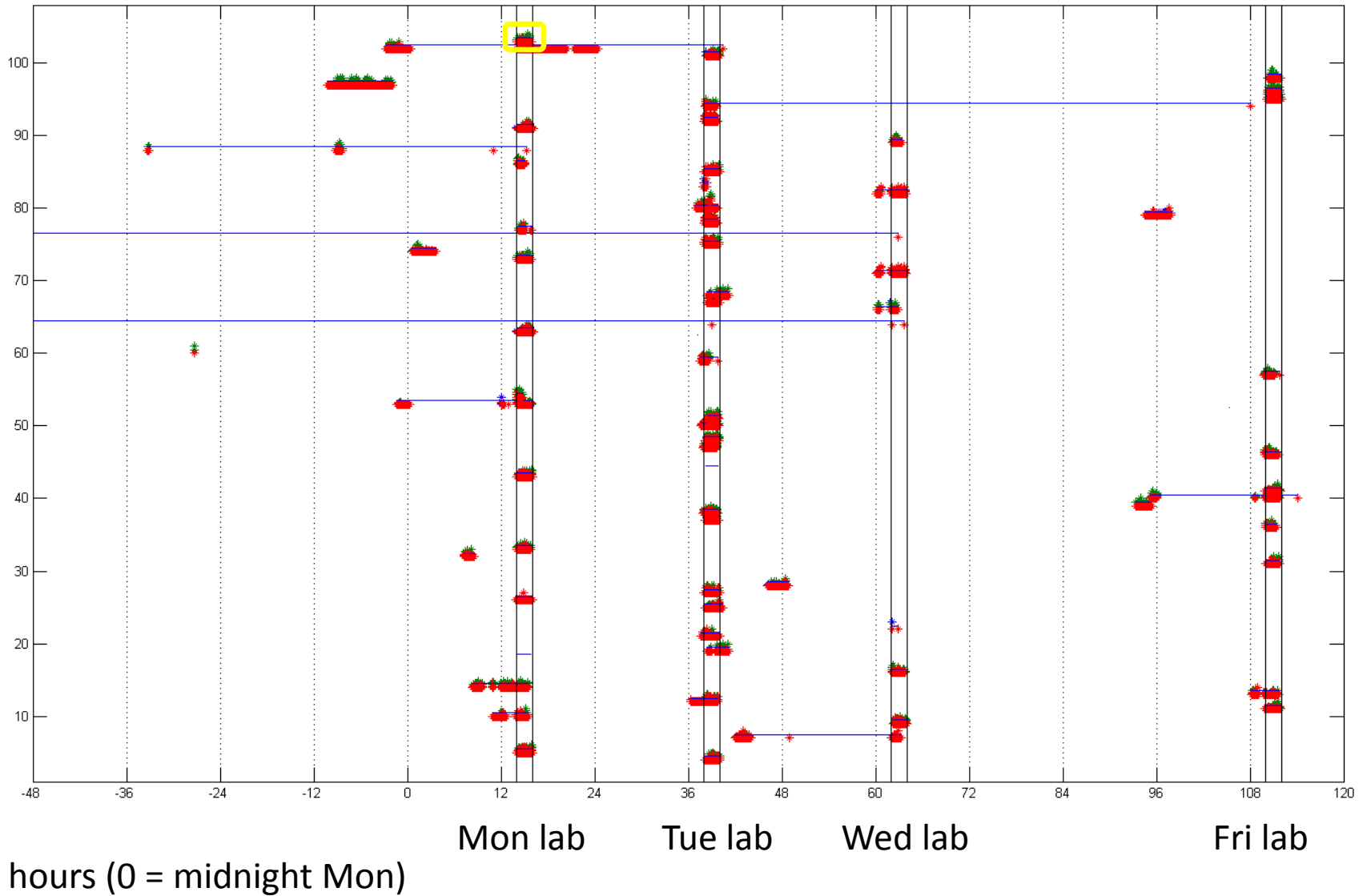
- How many students work on the activities prior to their lab time.
- Start times differ between labs on different days of the week.

Limitations:

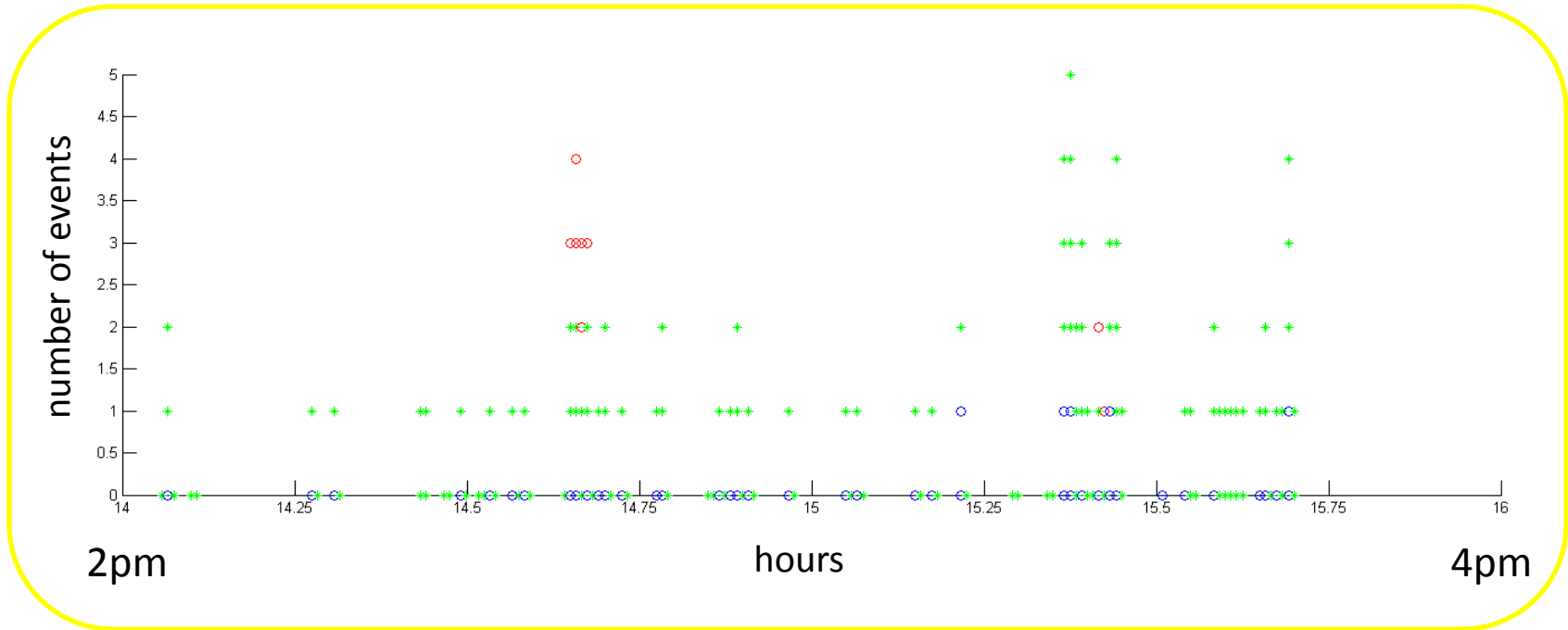
- Activity marks spread uniformly as there are no intermediate timestamps; students were asked to run a timestamp command but this was done unreliably.
- Not clear when breaks occur (class schedule indicates Wed students would have 9-10am block and then return at 2pm)



Trial 2: Automated Collection with Timestamps



Each horizontal line in the timeline above represents a student session. Expanded, the one in the yellow rectangle looks like:



Blue circles indicate file changes, green marks represent a run of the current program, and red circles indicate MATLAB interpreter errors (here, the errors clustered between 2:30pm and 2:45pm are all associated with plotting routines).

New Information

This more detailed system provides:

- Clearer picture of when work before assigned lab time occurs.
- Coverage of a wide range of students, including those who don't ask for help from the TA or others.
- Detail in the individual sessions; example session shows:
 - This student completed all the week's work during their lab session (no early start).
 - Minimal syntax errors.
 - Some issues with plotting routines.
 - Looking at file changes can indicate time on each activity.

Possible Applications

- Track time spent on specific activities, to help evaluate feasibility/difficulty level.
- Run during a programming “test”, short window of controlled time for a task.
- With a certain activity design, could implement an automated grading scheme on these sorts of log files.
- By adding an automated upload over the network, TA or instructor could monitor lab work in real time in order to provide useful feedback.

Conclusion

In-lab observations of student programming can be extremely useful in identifying the difficulties of novice programming on given tasks. This new MATLAB session logging scheme can complement such observations by unobtrusively (and in a highly automated way) providing more specific information about more students in their work.