DS Terms 2-3, Fall 2022

Course: *Numerical Analysis on a Pocket Calculator*

Discipline(s): Mathematics (Applied) and Computer Science

Instructor: Brian Hill

NB: I make the following mapping when applying the GODSAP template to science and math courses:

- Critical Thinking Skills → no change
- Reading Skills → Following Technical Arguments and Problem Solutions
- Speaking and Listening → Working Collectively to Understand Technical Material
- Writing Skills → Solving Technical Problems
- General Reflections → no change

Evaluation of *Numerical Analysis* with respect to the GODSAP

Critical Thinking Skills

There is a lovely art to working within the confines of a 1970s programmable calculator. It forces one to be clear about the theory and the abstract algorithms being applied, and then to express them concretely and economically. Understanding the theory of these algorithms and getting programs to operate correctly both sharpen critical thinking. Particularly important to critical thinking in the current policy milieu, where technocrats are now dominating the public's interaction with science, is the ability to understand the meaning and significance of statistical results. The third of the course's four major topics focused on that.

Following Technical Arguments and Problem Solutions

We surveyed a large variety of applied mathematical techniques. The course's four major topics were:

1. Operation and programming of a stack-based calculator, the Hewlett-Packard 25
2. General applications that were in the calculator's target market: games, finance, navigation, and surveying
3. Statistics: linear regression, exponential, logarithmic, and power law curve fitting, standard deviations and correlation coefficients (r-squared), t-test and χ-squared hypothesis tests
4. Numerical analysis: Newton's root-finding method, Euler's method for first-order differential equations, numerical integration

There is a rich variety of background needed to deeply understand these subjects. The classes required the students to understand this background sufficiently to use and program algorithms based on them.

Working Collectively to Understand Technical Material

There were some disruptions this semester. One student was half-in/half-out, and ended up dropping out of her second year at Deep Springs. We also had another bout of COVID which affected most of the student body about 60% of the way through the course. Despite these disruptions, and especially once they had passed, a productive and warm collective mode of inquiry emerged. Often that meant that the instructor gave a quick theoretical introduction to a topic, and then we worked together on examples and applications. Alternatively, it meant that the students learned the theory from our main source (the *HP-25 Applications Programs* book), and came in prepared to discuss and apply the theory and the programs.

During the final weeks (weeks 14 and 15), special projects were undertaken. This was a wide open opportunity to elaborate or apply any topic we had covered, or to strike out in a different direction. Again a productive and warm environment emerged.

Solving Technical Problems

We had eleven assignments, two exams, and lots of in-course problem-solving. On the positive side, we did a very substantial tour of many topics in applied mathematics. On the other hand, because we never spent more than about two weeks on a topic, I am concerned that the retention level will be modest. As an example of this, although the students learned and used both the t-test and χ-squared hypothesis testing, and demonstrated their understanding on assignments and exams, it may quickly be forgotten. In that case, I have to hope that they will be able to review what they learned and bring it back into their working memory.

With respect to the design of the HP-25 calculator, all of the students became pretty skilled with it, and did so with little or no resistance in the first half of the first term. This was pleasant and a little unexpected, because programming in other environments often gets bogged down in just having facility with syntax. Perhaps because the HP-25 is so well-designed and has limited goals and limited capacity, mastery of its capabilities was well within their reach. Or perhaps this was just a particularly skilled and agreeable group of students.

General Reflections

I have already mentioned that the tour of topics had both pluses and minuses, but as part of my general reflections, I want to focus on that, and then one other thing.

The pluses are that this was a very substantial tour, and it will serve the students well when it comes time to build on any of the topics. The minuses are that there wasn't a thematic development as the tour progressed, and this means the topic lacked cohesion.

The other thing that I haven't previously mentioned, is that too many of the problems we tackled were already well-solved in the HP-25 documentation. This meant that the opportunities for independent programming of the HP-25 were less than I had originally planned on. I am not sure how I would rectify this. It is simply a fact that the classic problem are already well-solved. Probably it means that we should have even more of a focus on meaningful applications, such as we had with the differential-equations solver.