

A Note on Reporting Programming Assignment Results

When reporting your results for a programming assignment the following format is recommended. It comprises seven sections each of which provide a different level of detail of the problem and its solution. Your goal in writing it is to demonstrate that you: understand the problem, identified and understand the relevant practical and theoretical aspects required for its solution, designed an efficient implementation of the codes required for the solution, and carefully designed a set of experiments that strongly support the correctness of your implementation. The sections and their purposes are described below.

I. Executive Summary

This should be a very short summary of the problem solved, the methods used, the results and the conclusions. It is not meant to give the details necessary for grading. For example,

A fourth order Taylor polynomial was used to approximate $\exp(x)$ on $[-1, 1]$. An analytical error estimate was derived. Computed results show the true error to be within 10% of the estimated error.

Another example:

A set of iterative methods for solving nonlinear equations were evaluated as to their effectiveness in determining the parameters in the proposed model for the data given. The resulting parameter values were guaranteed to be accurate to three decimal digits.

II. Statement of the Problem

A short summary of the problem to be solved. A few sentences is all that is needed here but it should provide more information than in the Executive Summary. You should not simply reproduce the paragraphs of the homework assignment. This summary shows that you understood the problem and can state it clearly and succinctly.

III. Description of the Mathematics

Briefly describe the mathematics used to formulate and solve the problem. You may of course consult the literature outside of the class notes. All use of material in these external resources and the class notes must be cited appropriately.

Some programming assignments will include specific questions that require explanation or analysis, e.g., proving a particular convergence rate. Answers to these questions should be included in this section but clearly labeled to indicate the portion of the assignment addressed. In some cases, these answers will

require verification experimentally, as discussed below, and pointers to those tests later in the document should be given.

IV. Description of the Algorithm and Implementation

Explain the methods used and how you implemented them. Of particular importance is discussing the aspects of the implementation related to efficiency, i.e., the number of computations and the amount of space required to solve the problem. This should include a description of relevant data structures or techniques used to keep the number of computations at an acceptable level by exploiting structure in the problem. If it is useful you may include short code segments from your solution to clarify the point to be made.

V. Description of the Experimental Design and Results

Describe the motivation behind the design of the experiments. This can be simple or require some significant insight depending on the problem. Recall that you have theory and empirical evidence covering what behavior should be observed for the algorithms we discuss. Use this to design experiments and evaluate relevant properties of the behavior of an algorithm, e.g., convergence rate, error estimates, etc.

The results should be organized and condensed. **Do not turn in pages of raw data and do not turn in a small set of examples that you claim “work”!** Since you will be running several problems, consider the careful use of summarizing statistics, such as means and variance, and tables, histograms and graphs. Be sure to give a complete description of the experiments and the conditions under which they were run. Predictions of the results and comparison with the observed behavior can be made in this section and discussed in a more global fashion in the Conclusions.

VI. Conclusions

This is the most important part. This section contains your structure discussion of the results. You should refer to your predictions, descriptions, and data from earlier sections.

Outliers in the data are often the indicators of problems with the codes. Make sure you explain them or remove them by fixing the bugs responsible. You may have made invalid assumptions in designing the experiments or in your expectations. A good explanation could save an otherwise useless set of incorrect results. The organization should reflect the fact that you are trying to convince the reader that your codes are correct and you understand the problem and its solution. It is important that you show an understanding of the problem, experiments and implementation. Correct answers are not the main or only point.

VII. Program Compilation and Execution Instructions, and Code Listing

In general, I will not be grading programming style beyond efficient coding that is described above. I may check your claims of efficient implementation versus the code actually used or have suggestions on coding techniques where necessary. A program listing should not be included directly in the report. Please submit the code in files that enable compilation and execution as well as any support files, e.g., makefiles. In Section VII of the report you should include descriptions of the files submitted and instructions to compile and execute the program(s) using these files.

Citations and Plagiarism

For some assignments it may be appropriate to modify existing codes from other sources or to use them in the testing process of your programs. Clearly, these codes cannot comprise your entire solution and regardless of source you must demonstrate mastery of the algorithmic, implementation and testing issues associated with the codes. **All such codes and supporting information must be cited clearly and their use specifically described. Failure to do so will result in a grade of 0 for the assignment.**