

Template for evaluating and grading projects, with grading scale based on achieved points

Evaluation of project number:

Name:

Abstract

Abstract: accurate and informative? Total number of possible points: 5

Mark and comments:

Introduction

Introduction: status of problem and the major objectives. Total number of possible points: 10

Mark and comments:

Formalism

Formalism/methods: Discussion of the methods used and their basis/suitability. Total number of possible points 20

Mark and comments:

Code, implementation and testing

Code/Implementations/test: Readability of code, implementation, testing and discussion of benchmarks. Total number of possible points 20

Mark and comments:

Analysis

Analysis: of results and the effectiveness of their selection and presentation. Are the results well understood and discussed? Total number of possible points: 20

Mark and comments:

Conclusions

Conclusions, discussions and critical comments: on what was learned about the method used and on the results obtained. Possible directions and future improvements? Total number of possible points: 10

Mark and comments:

Overall presentation:

Clarity of figures, tables, algorithms and overall presentation. Too much or too little? Total number of possible points: 10

Mark and comments:

Referencing

Referencing: relevant works cited accurately? Total number of possible points 5

Mark and comments:

Overall

Overall mark in points (maximum number of points per project is 100) and final possible final comments

Grading of all projects

The final number of points is based on the average of all projects (including eventual additional points) and the grade follows the following table:

- 92-100 points: A
- 77-91 points: B
- 58-76 points: C
- 46-57 points: D
- 40-45 points: E
- 0-39 points: F-failed

General guidelines on how to write a report

Some basic ingredients for a successful numerical project

When building up a numerical project there are several elements you should think of, amongst these we take the liberty of mentioning the following:

- How to structure a code in terms of functions
- How to make a module
- How to read input data flexibly from the command line
- How to create graphical/web user interfaces
- How to write unit tests (test functions)
- How to refactor code in terms of classes (instead of functions only), in our case you think of a system and a solver class
- How to conduct and automate large-scale numerical experiments
- How to write scientific reports in various formats (LaTeX, HTML)

The conventions and techniques outlined here will save you a lot of time when you incrementally extend software over time from simpler to more complicated problems. In particular, you will benefit from many good habits:

- New code is added in a modular fashion to a library (modules)
- Programs are run through convenient user interfaces
- It takes one quick command to let all your code undergo heavy testing
- Tedious manual work with running programs is automated,
- Your scientific investigations are reproducible, scientific reports with top quality typesetting are produced both for paper and electronic devices.

The report: how to write a good scientific/technical report

What should it contain? A typical structure

- An abstract where you give the main summary of your work
- An introduction where you explain the aims and rationale for the physics case and what you have done. At the end of the introduction you should give a brief summary of the structure of the report
- Theoretical models and technicalities. This is the methods section
- Results and discussion
- Conclusions and perspectives
- Appendix with extra material
- Bibliography

Keep always a good log of what you do.

The report, the abstract

The abstract gives the reader a quick overview of what has been done and the most important results. Try to be to the point and state your main findings.

The report, the introduction

When you write the introduction you could focus on the following aspects

- Motivate the reader, the first part of the introduction gives always a motivation and tries to give the overarching ideas
- What I have done
- The structure of the report, how it is organized etc

The report, discussion of methods, implementation, codes etc

- Describe the methods and algorithms
- You need to explain how you implemented the methods and also say something about the structure of your algorithm and present some parts of your code
- You should plug in some calculations to demonstrate your code, such as selected runs used to validate and verify your results. The latter is extremely important!! A reader needs to understand that your code reproduces selected benchmarks and reproduces previous results, either numerical and/or well-known closed form expressions.

The report, results part

- Present your results
- Give a critical discussion of your work and place it in the correct context.
- Relate your work to other calculations/studies
- An eventual reader should be able to reproduce your calculations if she/he wants to do so. All input variables should be properly explained.
- Make sure that figures and tables should contain enough information in their captions, axis labels etc so that an eventual reader can gain a first impression of your work by studying figures and tables only.

The report, conclusions and perspectives

- State your main findings and interpretations
- Try as far as possible to present perspectives for future work
- Try to discuss the pros and cons of the methods and possible improvements

The report, appendices

- Additional calculations used to validate the codes
- Selected calculations, these can be listed with few comments
- Listing of the code if you feel this is necessary

You can consider moving parts of the material from the methods section to the appendix. You can also place additional material on your webpage or GitHub page..

The report, references

- Give always references to material you base your work on, either scientific articles/reports or books.
- Refer to articles as: name(s) of author(s), journal, volume (boldfaced), page and year in parenthesis.
- Refer to books as: name(s) of author(s), title of book, publisher, place and year, eventual page numbers