Chapter 1

Introduction

1.1 Motivation

The aim of this course is to teach you technical writing in the mathematical sciences. Knowing how to write high-quality technical reports is of practical relevance whether you want to become a researcher or work in industry.

There are two parts to writing a technical report, doing research work worthy of being reported, and then preparing the actual report itself. In this course you will do both. You will work on four independent projects and will then write technical reports on the findings you obtained.

The four projects are practical and contain aspects you most likely will encounter in practice in your career after graduating. These projects are designed to challenge you, in that they introduce problems that you may have never seen before. They will introduce mathematical concepts you most likely will have never seen before. While this may seem intimidating, solving new problems and being able to figure out how to use mathematical and computational tools you have never used before is probably your most valuable skill as a mathematician. Employers will hire you because you possess this skill.

Working on problems that go beyond your current mathematical understanding is an important exercise you should aim to practice continuously during your undergraduate program. This will allow you to gain experience that is immensely valuable both for academia (should you plan to go to graduate school) and private-sector industry.

To guide your exploration on the projects to be tackled in this course, for most projects a concise introduction to these problems will be provided. This introduction should be enough to get your started but may not go deep enough for your personal taste. Therefore, some key references will be provided where further information could be found. Those references should mostly be regarded as a starting point for your own literature review, they may not be the references that will be most helpful for everyone. Some students may prefer classical textbooks, other may prefer blog posts on the internet explaining the topics covered in these projects. Learning to do a proper literature search is an important part of the scientific writing experience.

Regardless of the problem you are working on, changes are high that someone else has worked on the exact same problem before your. Chances are even higher that someone else has worked a similar problem and used methods that will be helpful for your specific problem. As such it is best practice in scientific research to begin with a literature review.

There are many ways how a scientific literature research can be conducted but one standard tool that is frequently used is *Google Scholar*,

https://scholar.google.com/

where you can search for scientific papers and books on every subject imaginable.

1.2 Resources

For this course, no previous knowledge of LaTeX or `Python` is assumed. Below I list some of the resources that should help you getting started with both.

1.2.1 LaTeX

There are many great tutorials on LaTeX available online and you are encouraged to find a tutorial of your liking. A simple tutorial to start, which should be sufficient for most of what is covered in this course would be *Learn LaTeX in 30 minutes*:

https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes,

which is available on the website of the online LaTeX editor Overleaf. A more detailed introduction is Chapter 3 of the MATH 2130 manual available on MUN's Math&Stats department website:

https://www.math.mun.ca/~m2130/Manual/ch3_typesetting.pdf.

To use LaTeX you first have to either install it locally on your computer, with versions available for Linux, Windows and MacOS, or use an online service like Overleaf, which is free for single user. For installation guides see:

https://www.latex-project.org/get/.

While installing LaTeX is generally straightforward, should you run into any issues related to your specific hardware that cannot be resolved easily, I would suggest using Overleaf instead. Note that if you install LaTeX on your own computer you may also have to install a suitable LaTeX editor.

1.2.2 `Python`

`Python` is a high-level, interpretative programming language that has emerged as one of the standard tools in scientific computing, data analysis and machine learning. While for the purpose of this course we will use `Python` (version 3) mainly for simple short programs and visualization purposes, I strongly suggest that you make every effort to pick up as much of this language as you can. Most of the jobs available for applied mathematician in industry require solid experience in `Python` and the more often you use it for the various courses in your undergraduate program, the more experience you will gain using it.

As with LaTeX there are a multitude of tutorials and online resources (including free online courses, for example on YouTube) available that will allow you to familiarize yourself with the basic functioning of `Python`. One short resource to start with would be *A Byte of Python*:

https://python.swaroopch.com/.

There are a great many packages available which considerably extend the base functionality of `Python`. In the applied mathematics, scientific computing and data science setting, the most important packages to get familiar with are *Numpy*, *Scipy*, *Pandas*, *Matplotlib*, *Plotly* and *Scikit-Learn*. In this course we will use only some elementary aspects of these packages, so it will be enough if you familiarize yourself with the main purpose of these extensions rather than trying to understand them exhaustively (which would be an overwhelming undertaking).

`Python` can be installed on Linux, Windows and MacOS. As with LaTeX there are also online `Python` interpreter available. One particularly simple and convenient way to use `Python` is via *jupyter notebooks*:

https://jupyter.org/.

`Jupyter notebooks` allow mixing text with code and thus allow for a neat and interactive way to run and present `Python` code and its associated results. I strongly advise to use `jupyter notebooks` for this course as a means to provide the code for the projects to accompany your written reports.

Personally I use `Google Colab` (which is free):

https://colab.research.google.com/,

for most of my `Python` programming, which combines the convenience of `jupyter notebooks` with the ability to interact with local files saved on Google Drive, exporting notebooks to `Github`, etc. It also comes with essentially all `Python` extensions pre-installed that are needed to do most projects in applied mathematics, scientific computing and data science.

Should you run into any issues installing `Python` or any of the associated packages on your local computer (which is fairly straightforward on Linux and MacOS, but slightly less straightforward on Windows), or if you encounter severe performance issues (maybe due to having an outdated computer), then I strongly advise to use an online `Python` interpreter such as `Google Colab` instead.

1.3 Style guidelines for your reports

A template for a report (along with the LaTeX source code) can be found on the course website on Brightspace. It follows the typical style of essentially all papers in the mathematical sciences:

1. *Title*: Choose a meaningful (but typically short) title that gives the reader a reason to look at your work.
2. *Authors:* The list of authors who contributed to the paper in an essential way, along with their work addresses and contact information (for you this is just your own name, since all projects are to be completed by yourself).
3. *Abstract*: This is a very short summary of the main findings of your work. The abstract would normally be the first part read by a prospective reader so be sure to make it

advertising and easily readable for the reader to be keen on reading your paper in its entirety.

4. *Introduction*: The introduction would usually contain a broader, mostly non-technical discussion of the problem you are considering in your paper. This is typically where you would include the literature review for your work.
5. *Methods*: Here you would begin to delve into the technical details of your work, e.g. explaining the mathematical or computational methods used.
6. *Results*: Applying the methods described in the previous section, this would be the place to include your results, including tables, figures, etc.
7. *Conclusion*: Here you would summarize the results of your paper and could provide some outlook of what could be considered as next steps in the wider context of the work in which your paper is situated.
8. *Acknowledgements*: Here you would thank people that have helped you in some way with your work (e.g. any peer-reviewers, colleagues, etc.), and acknowledge funding from sponsors.
9. *Appendices*: This is optional, but could include the source codes used in your work, longer proofs to theorems, etc.

The above is a general formula, and not all papers have to follow the exact same formula. Shorter papers may combine the *Methods* and *Results* section, and this may also be appropriate for some of your own reports for this course. Many papers within the area of pure math also do not separate the *Methods* and the *Results* sections, since the main aim of such papers is to prove one or more statements so there is no natural separation between the methods used and the results obtained.

Besides the above general structure for a mathematical report, the following is a short collection of best practices that you should keep in mind when writing your reports:

1. Technical reports are typically written in neutral, concise language; exaggerations should be avoided.
2. Technical reports are more often written in Third Person ("We show that") rather than in First Person ("I show that"), even if there is only a single author.
3. Despite the language in technical reports being neutral, is has to follow all the regular rules of the English language.
4. Your task is to back up everything you are reporting. Sweeping statements without proofs have to be avoided.
5. If you have a conjecture about a statement that you cannot back up, it is acceptable to include this as a conjecture, and explain your reasoning behind it, and what would be required to exhaustively prove this statement.
6. Including plots and tables can make a report more easily digestible, providing summarizing information in an accessible way. For this to work, plots and tables should be as self-contained as possible: Axes have to be labelled; multiple curves should have a legend; if plots are in color, think if they would be still understandable if printed in black and white; if numbers are reported then explain the units being used.
7. If you use information taken from books or scientific articles it is imperative to cite them.

As with any kind of writing, also scientific writing requires a lot of practice. Do not get discouraged if you find it difficult or overwhelming to getting started! For more suggestions, consult the MATH 2130 course manual or [2].

1.4 References

A hallmark of a technical report is that is includes references to other technical documents, such as papers, technical reports or books. As you research a scientific problem, you would start with an extensive literature review. There is not much worse for a scientist than to write a paper on a topic only to discover later that the exact same problem has already been treated elsewhere. To avoid such an unfortunate situation, knowing your subject area is key.

While for the present course some background information on the problems to be considered is already given in these lecture notes, a good report will expand on the problems you will be working on. Doing further literature research, reading papers and appropriate sections of books will strengthen the report you will be writing.

It is crucial to stress the importance of proper citations of sources used. If you find a useful statement for your report in a paper, then you have to cite that paper, and make it part of the bibliography of your report. LaTeX provides a suitable reference management system in the form of `BibTeX`, which allows you to easily add references into a database which can then be used within LaTeX.

There are many different citation styles, but the most common in the mathematical sciences is to either cite a paper by its reference number of the bibliography (as is done in these lecture notes) or in the form of Authors/Year. To cite the book by S.G. Krantz you thus could either use the numerical form, that is [2], or the Author/Year form, that is (Krantz, 2017). The style of your bibliography can be set globally in your LaTeX document. The ordering of the bibliography can be customized as well, with alphabetic ordering being the most common (another option would be to order references according to their occurrence in your report, meaning the first paper you cite would be [1] in your bibliography, the second paper [2], etc.).

Independent of the citation style you are using, references are important. It is not a weakness to cite many papers in a report, as long as the references you cite are relevant to your work. In fact, unless a statement is within the domain of common knowledge (such as the Pythagorean Theorem, the rules of calculus, etc.), it has to be cited.

The source from which you cite is also important. Books and peer-reviewed scientific papers are more credible sources than a blog post off the internet, which usually has not been peer-reviewed and which may be taken down at any point. It is of course absolutely allowed to use information from the internet and cite it accordingly (just indicated what date you accessed that information), but actual research papers largely avoid material from the internet, simply because most of the scientifically useful material can be found in more credible sources. Please consider this for your own report and avoid using pages off the internet as your sole source of information.

As an example, when you are working on Project 2, and you want to describe the Lorenz model in some more detail, don't cite the Wikipedia article on the Lorenz model. The original paper by [3] would be a great source. If that information is too hard to digest for you, try finding an introductory book on dynamical systems, chaos theory, predictability, etc, which shortly describes the Lorenz system and provides some useful background information for you to include in your report. According to Google Scholar, the paper by Lorenz has as of August

2020 been cited 22634 (!) times, so there is an abundance of information on this system out there that will be accessible to every skill level.

The same goes for the other projects as well. I have provided some suggested reading for Projects 3 and 4 as well, but please consider them a starting point for your own literature research rather than the end.

Using information without properly citing the relevant sources (independent of whether these sources are books, articles or the internet) is plagiarism, which is a serious academic offence, both in science and for this course. Avoid plagiarism at any cost!

1.5 What to include in your report

In Section 1.3 I have provided a short possible skeleton for your report. What precisely to include in each section is largely up to you. Here are some general thoughts you might want to consider:

1. *Introduction:* A good report should be self-contained, i.e. it should be readable by a colleague who has a solid mathematical education but who would not necessarily be an expert in the specifics of the report. For Project 1 this would mean that you could present some background on climate change and its impact, for Project 2 you could provide an introduction to (numerical) weather forecasting and how it relates to ensemble forecasting, etc. Your goal for the introduction would be to convince the reader that what you are presenting in your report is an important problem that needs to be studied further.
2. *Methods:* Here you would explain why you chose the methods you were using. You would explain the model you were considering in such a manner that it would be readable to someone who has never seen that model before. While for the purpose of this course the models and methods are mostly given (e.g. numerically solving the Lorenz system for Project 2), you would still provide arguments as to why this is the right choice. You could also discuss competing or more general approaches (as applicable), highlight potential limitations, explain why you believe your approach was appropriate, or how these limitations would restrict the generality of what you were doing.
3. *Results:* Here you present a selection of your graphs, tables, etc. An interpretation to all these results has to be given. That is, it is not enough to just include the plots you were asked to produce without any further explanation. Give a critical analysis, explain what these plots are showing, how they are solving the problems you set out to solve, where they fall short, etc.
4. *Conclusion:* Give a concise summary of what you have accomplished in your research and why it is relevant. Honestly assess the strengths and weaknesses, e.g. where further research would be needed to corroborate your hypotheses. Shortly describe what you think would be appropriate to do next within the wider area of that project. Would more complicated models have to be considered? Would you need more data to make more justified statements about that research problem?
5. *Appendix:* Here you could provide (parts of) the `Python` codes you have written. If your code is very long then it would not be appropriate to include all of it as this will make your report appear rather messy. In particular, the various `import` statements, variable initializations, etc. can usually be omitted; rather, you could select and discuss some of the key routines you have written, in particular if they clarify other parts of your report.

In other words, if you describe your methods in the *Methods* section, you could reference your computational routines in the *Appendix* for clarification purposes.

There is no general rule as to how to structure a research paper, and how much weight to assign to each section. Different colleagues will have different opinions, and ideally you will find your own style that will work for you. Aim for a paper that is interesting to read, factually correct, and that does a proper job in convincing the reader that what you have done is of scientific value and should be considered further.

The goal of research is to produce papers that will be read and cited by colleagues. Writing a paper that nobody wants to cite is frustrating for the authors. While most research is highly specialized and will not gather as many citations as the aforementioned paper by Lorenz, it is still the case that the presentation of your research results is a main contributor of how your paper will be received by the scientific community. You may have proven an important statement or obtained an important result but if you present the proof or that result in an incomprehensible or sloppy way, riddled with typos and grammatical mistakes, then chances are high that your paper will not be successful.

Please remember this for your reports as well, *the presentation of your results is just as important as the results themselves.*

Remark 1. The length of a report is **not** an indication for the quality of a report. A concise, well-thought out 2 page report will be better than an unstructured and unorganized 10 page report. As everybody has a different style, I will not provide guidelines on the lengths of your reports!

Remark 2. In computational mathematics it fortunately becomes more and more customary to provide the source codes for the research you have done. This was unfortunately not the case in the past, which made it hard for reviewers to assess the correctness of the results reported in a scientific paper. The source code can be provided in various ways, e.g. making it publicly available in repositories on online code hosting services such as `Github`, on the website of the journal, or in the appendix of your paper itself. Here we will follow best practices so I ask you to submit your codes along with your reports. Short codes can be provided in the appendix of your paper, longer codes could be uploaded as `.py` (`Python`) or `.ipynb` (`Jupyter notebook`) files to the respective project Dropbox.

1.6 Submitting your report

When you try to publish a scientific paper that you have written in a scientific journal, it will have to undergo *peer-review*. Here the editor of the journal you are submitting your work to will select a few experts in the field of the article and will ask them to carefully read your paper and provide reports on it. Based on these reports, your paper will either be *accepted*, has to *undergo a revision*, or will be *rejected*.

We will use the peer-reviewed method for assessing your reports as well. Once you are happy with your report (or, at the latest, at the deadline for each project) you will send it to me as the ‘editor’ (in practice you just upload it to the assignment Dropbox on Brightspace). I will then send your report to two of your colleagues and ask them to write a short critique on your report. This critique should honestly (but politely!) assess the strengths and weaknesses of your report. Note that peer-reviews for journals are anonymous, so you will not know who will be reviewing your work, and the reviewers should not include their names in their reports.

Learning to write a report as a reviewer for a scientific document is an important skill as well. In practice you come across a variety of reviewers and not all of them are friendly and polite, and unfortunately not all of their reports are really useful for you as author at all. Here we aim to learn best practices of being a supportive reviewer, with the goal of improving your peers' reports. There will be no *rejection* option, but your goal as reviewer will be to find as many weaknesses as possible in the report you are reviewing, along with concrete suggestions for improvements.

As a reviewer, you can also go through the list of best practices provided in Section 1.3. Have these best practices been followed? If not, then you could provide some helpful suggestions on how the report could be improved. Are the results faulty? Are the arguments hand-wavy? Are the conclusions justified? Is the presentation of results understandable? Is enough background information provided?

Once your review report is done, you will send it back to me (again, there will be a Dropbox where you can upload the report), and I will then forward this report to the author of the paper you reviewed. The task of you as the author is then to incorporate the feedback you have received. You would correct any mistakes found by the reviewers, or, if you do not agree with a reviewer on some of his/her remarks, you would provide an argumentation as to why you did not incorporate these remarks.

The correction process thus consists of two steps: You correct your paper according to the suggestions of the reviewers, and you collect all of your corrections in a response document (usually entitled *Response to the reviewers*). To give an example, say your reviewer remarks that you forgot to label some axes in your plots, you would then (i) make new plots with the proper axes labels for your paper itself, and (ii) write in your response document that you have included these new plots. Practically, this could be done by copying the respective remark from the reviewer's report in your response document and providing your response thereafter, e.g.:

Remark by reviewer: "I should also like to note that in Figure 2 the x-axis has not been labelled."

Response: "We thank the reviewer for catching this issue. We have added a proper label to this Figure. The x-axis now correctly identifies this variable as *time*."

If there are multiple issues being raised by the reviewer it is not necessary to thank them for each and every single point. Still, try to maintain an overall grateful tone in your response document, even if you disagree with what the reviewer has been proposing. Staying polite despite having an unfriendly reviewer is a skill that unfortunately has to be honed in science (as well as in industry).

In practice, the editor would then forward your response document along with the corrected version of your paper to the same reviewers again, who then will make a final decision (or require some more modifications). While it is generally the case that there will be only one revision, some reviewers may require multiple back-and-forth until they will come to a final decision on whether your paper can be accepted or has to be rejected.

Here we will not do multiple rounds of revision. You will incorporate the feedback of your reviewers within **one week** and send the final version of your paper to me (in practice you will re-upload it to the Dropbox, along with your response document), and I will not send it back to the reviewers. This once-revised version of the paper is what I will then be grading.

As with writing a technical paper itself, also writing reviews and understanding the intricacies of the peer-review process is a skill that takes some time and practice to acquire; upon completing this course, you will have a better understanding of writing reports, reviewing and revising them.