Chapter 2

Technical writing

2.1 Technical versus non-technical writing

Most things about writing that you learn in English courses apply equally to technical writing. This chapter does not pretend to teach you the rules of grammar, basic principles of composition and style in general. We rather focus on the elements of writing specific to this course. (Yet some common spelling errors etc. will be mentioned — see Sect. 2.4.1.)

Writing requires you to organize your flow of thought into a coherent sequence of units carrying sense. The smallest such unit is a sentence. Then comes a paragraph. A short story may contain no further structural units, while more sizeable pieces of fiction are often organized into Parts and Chapters.

Scholarly writing, as compared to fiction, is characterized by a more sophisticated **hierarchy** of logical units. Some of them, such as Sections, Subsections, References, determine the **plan** of the paper. Others, such as Definitions, Remarks, Tables, help the readers to **pause and digest** one relatively small piece of information at a time. Good graphics can be truly informative and replace a hundred words. Particular to mathematical writing are such units as Theorems and their Proofs. Common in technical reports are fragments of computer code or whole program listings. In Section 2.3 we make recommendations concerning the global structure of your Math 2130 reports. Some useful advice can be found in Appendix B-1, Section 2.

The use of language in technical writing is strongly biased towards **precision and clarity** as opposed to figurative, metaphoric language common in non-technical narration. Technical writers should, as a rule, keep a neutral tone and abstain from emotional bursts. There are also specific problems: whether to use “I” or “We” or write in the third person; what tense to use, etc. Questions of this kind are discussed in Sect. 2.4 and B-1.3.

Writing mathematics properly is a special art, which does not come easily. Section 4 in Appendix B-1 should help you get started.

© MMIX Department of Mathematics and Statistics, Memorial University of Newfoundland
September 4, 2009

2.2 Writing process

Before setting out to write a report, one must have *something* to report. This means, in our case, *results*: a solution to the assigned problem. Obtaining the results usually takes up most of the time students spend on their Math-2130 assignments. Remember, however: properly presenting your work in writing is not a simple task, either.

With the results obtained and the pertinent information collected, do you need anything else before starting to write? Perhaps. There are questions to answer and decisions to make:

- What is the main **purpose** of your paper? Is it to illustrate a certain mathematical theory with examples you have produced (by hand and with the help of a computer)? Or is it to report your algorithm of solution of the assigned problem and the results of computations? Is it to decipher a cryptic message? Or is it, possibly, to describe a series of computer-assisted experiments with a logical game? Knowing the purpose will prevent you from going off on tangents when writing.

- Decide about scope. **Narrow down** the subject so as to avoid excessive generality. Decide what theory, which results, how many graphs and tables you want to include. Keep focused when writing. It is better to present a small circle of ideas accurately and precisely than to attempt to embrace a larger area in vague terms.

- Decide who is your assumed **readership** is. In most cases the best assumption is that the reader is a fellow student with a background like yours. In some projects, especially those of an entertaining nature, younger math students can be the target audience. Stick with the interests and level of your readers. Do not go over their head, but do not use baby talk.

  Aiming your paper at a graduate level audience or professors will only work in exceptional cases. Never address the paper directly to your professor as if he/she were the only reader.

In summary:

Identify the purpose. Limit the scope. Speak to your audience.

Now we come to the main point — how to actually put things down. Writing is a creative process. One's writing strategy reflects one's personality. There is no such thing as a magic writing "algorithm" suitable for everybody. Some people write easily; most struggle. Find what works best for you, making writing less painful and more enjoyable. As food for thought, here are several possible approaches that different people use. Your writing strategy will likely be a mix of these, or possibly even entirely different.

- *Top-down approach*: Start with a plan, set the goal or goals, and proceed section by section. This approach, generally speaking, requires good self-discipline and ability to comprehend the material of the paper in its entirety; otherwise it is easy to miss important points at the outset. The suggested format of a Math-2130 paper (Section 2.3) will hopefully help. You will still need to return to sections already written and to make changes as your work progresses.

- *Bottom-up approach*: Begin by stating the results, then trace back. A method by which the results were obtained must be fully explained. Terminology involved must be defined. Elaborate, fill in details. Make sure the final order of parts makes logical sense: concepts should be introduced before they are referred to.
- *Free flow approach*: Start writing up your thoughts as they come. Write an introduction. Put down relevant definitions, facts, considerations. Describe the work done and the results obtained. State a conclusion. Then edit the obtained loose draft: organize the material into structural units, improve explanations (expand where needed), remove redundancy. In the end, rewrite the introduction anew.

No matter what approach, the chance of writing a good paper from scratch in one attempt is small. Editing will be necessary. Read what you have written. Note what sounds ugly, awkward, cumbersome. Strive for clarity. Move material around to achieve the best logical order. Replace vague phrases and words with clear-cut ones. Check your writing against this Manual; cross-read the papers with a friend; consult with the Writing Centre.

2.3 Organization of report

2.3.1 General requirements

A typical report in this course contains or may contain the following components, in this order:

- Title page
- ○ Table of contents
- Abstract
- Body of report:
    - Introduction
    - Technical Details
        - Mathematical details
        - Program details
    - Results and Analysis
    - Conclusion
- ○ Acknowledgements
- ○ References
- ○ Appendix

The items marked with solid bullets are mandatory; the presence of others depends on the circumstances. We'll comment on each of them, one by one.

The body of your report (from Introduction to Conclusion), excluding in-text graphs and illustrations, should not exceed **six printed pages**. This means that you must clarify your ideas and arguments and write them in a very precise and concise way.

2.3.2 Title page

The title page should give the following information:

- The title and number of the lab.
- The course name and number (Applied Mathematics 2130).
- Your name and student number.
- Your professor's name.
- The date of submission.

A typical title page is shown in Figure 3.2 in Section 3.2.1, where the LaTeX code used to produce it is also given.

About the **title**: try not to repeat the title of the assignment. Devise your own way to describe the topic. The title should not be too general. For example, *Solving a Mathematical Model by Means of Computer Programming* isn't good (although it can be a title of choice for an introductory lecture in a Math Modeling course). While being short, the title should reflect particulars of the work. For example, *Two methods for evaluating the volume of a pyramid* is much better than just *Pyramids.* Word play, subtle humor, puns — these may work, but make sure you show a good taste. Some variants are totally inappropriate: like *Mathematics Supporting Global Warming.* Put *Behind* in place of *Supporting* — and the title becomes acceptable. A perfectly normal, if not at all fancy, version is *Mathematical Modeling of ...*.

2.3.3 Table of contents

The Table of contents is generated automatically by LaTeX from the headings of your sections. Some extra commands are needed in order to include the References and Appendix sections. The details are given in Sect. 3.2.2.

A safe practice, in the first assignment at least, is to adhere to the suggested standard plan and headings. Then, as your experience grows, you can vary the paper structure and the headers of sections and subsections. For instance, the title *Mathematical details* can be changed into *Geometry of tangent circles*, if that's the mathematical subject dealt with. It can be further divided, if appropriate, into subsections like *Tangency condition in terms of coordinates*, *Relation between the radii of four tangent circles*, etc. The table of contents that exhibits such subject-specific and self-explanatory headers allows the reader to grasp the developments in the paper at a glance.

In short reports, the table of contents may not be needed at all. You may still be asked to include it as a typesetting exercise. Check with your instructor.

2.3.4 Abstract

From a reader's perspective, the abstract serves the purpose of classification: where does the paper belong? Should I take a closer look? The abstract should be short yet informative. Refer to the real-world problem or the mathematical question that gives rise to the project. Define the area of mathematics involved. Briefly characterize an algorithm and/or a program. Squeeze in the essence of results or mention a particularly striking result, perhaps schematically, in an easy-to-grasp form. Avoid extensive details.

In practice, abstracts are often restricted to a prescribed numbers words. Try to limit yours to 100 words. Learn word-saving tricks. Cross out epithets, excessive verbiage, and the obvious. There is absolutely no room for duplication, elaboration, and emotions.

Example. This is a bad abstract:

> **Abstract** In this paper, we discuss how to create a program that is useful for modeling global warming. Unlike in the real world, however, where water vapor and carbon dioxide both play an important role, our program makes simplifying assumption that there is only one gas responsible for the greenhouse effect, whose concentration is proportional to re-radiation of heat. Finally, results of simulations are presented.

Cut out deadwood and unnecessary elaboration, preserve the useful particulars, add some specifics on the method and results, — and a much better version is obtained:

> **Abstract** We discuss a computer simulation of global warming based on a simple mathematical model, in which one gas is responsible for the greenhouse effect and its concentration is proportional to re-radiation of heat. The program iterates over time steps, one per season. Instability of temperature is observed if the intensity of gas emission exceeds a certain critical value.

2.3.5 Introduction and Conclusion

If after glancing at the Abstract the readers feel the paper is not out of touch with their interests, they browse through the Introduction and Conclusion. The central part of the paper, with all the technicalities, is the last place the readers will go.

The **Introduction**, as the name suggests, should introduce the reader to the problem being investigated. Motivation and historical background can be included (although some of it can be scattered over later sections, too). The Introduction should also indicate what the reader will find in the remainder of the report. The context and language of the Introduction often makes it clear who the target audience is. Otherwise, state any special assumptions about the readers' background explicitly, e.g. "We assume the reader is familiar with eigenvalue theory for matrices."

When writing the Introduction, assume that the original assignment is not available to the reader. Your paper must be self-contained. Do not copy the assignment's language; use your own words. Some assignments might introduce a little story and characters, like Alice and Bob. In this case, again, your own Introduction must independently describe the situation — so that the reader who didn't see the assignment sheet would know what you are talking about.

The introduction can be viewed as an extended abstract, but it has a broader mission. Hook the readers; make a promise that makes them want to stay with your paper. A potential reader may never get to appreciate the rich and interesting contents if the introduction fails in its mission.

For a typical paper in this course, the Introduction should be from 1/4 of a page to a full page long. It should not tire the reader. It should be rather easy reading, not so technically dense as the subsequent sections. In many cases it is not a place to put precise definitions, but rather a place to motivate and anticipate them by describing the major concepts in a less formal way.

Example 1.

> The notion of two graphs having *identical shape* may be important. What exactly does it mean for two graphs to have an identical shape? There are geometric definitions and analytic definitions. They will be discussed in Section 2.

Example 2.

> When we say that one geometric figure is an *expansion* of another, there is an intuitive understanding of the two being alike and differing only in size. For the purposes of this project, a precise geometric definition is required. It refers, in turn, to the notion of *isometry*, or distance-preserving transformation of a plane, and to the notion of *homothety*, which is stretching or squeezing in the same proportion along all directions going from a fixed center. [Then the introduction proceeds informally. In a later section, the technical definitions, say, in a coordinate form, are given.] ,

In some cases a precise technical definition properly belongs in the introduction. Suppose the assignment asks you to write a program that counts in how many ways a given positive integer N can be broken into a sum of positive integers. It is rather pointless in this case to keep an informal tone. Get straight to the point:

> The purpose of this laboratory is to design a method and to write a computer program for counting partitions of integers.
>
> **Definition**. A *partition* of a positive integer N is a set of positive integers arranged in non-increasing order $n_1 \geq n_2 \geq \ldots \geq n_k$ such that $n_1 + n_2 + \ldots + n_k = N$.

You can help the reader to understand the definition without breaking away from the formal style by adding a comment or remark explaining important particular cases. For example: "A partition with the least value $k = 1$ consists of one element $n_1 = N$. A partition with maximum number of elements $k = N$ is $n_1 = n_2 = \ldots = n_N = 1$". The project may later deal with special classes of partitions, say, those with non-equal members. It is not necessary (and hardly appropriate) to put all definitions in the Introduction.

The last section — **Conclusion**, or Concluding Remarks — contains a brief **summary** of the findings of your report. By reading only the summary, a reader should be able to ascertain the most important facts resulting from your work. Do not overload the conclusion with details of the results.

It is tempting to create the Conclusion from the Introduction by a simple "copy and paste" method. However such an approach misses the point. Observe the difference. If your introduction sets up a goal or makes a promise (as we suggest it should), the conclusion serves as a "checklist": has the goal been reached? is the promise fulfilled? Sometimes the answer will be — not quite; in that case, you should admit it and explain.

Example.

> In this paper, a method for counting all partitions of a given positive integer N is described. A FORTRAN program has been written and the number of partitions has been computed for all $N \leq 30$. It is apparent that the number of partitions $P(N)$ exhibits a rather rapid growth as N increases. We observed and proved that $P(N) > N^2$ for $N > 9$ and that $P(N) < 2^N$ for all N, but we did not come up with a definite conclusion about the precise law describing the growth.
>
> The program presented here uses direct enumeration of partitions. A Wikipedia article [2] suggests another, supposedly more economical method for counting partitions, based on recurrence relations. We have also attempted to implement that method but have not been able to complete the programming in a timely manner.

It is too late in the conclusion to bring in new material not found earlier in the paper. Instead, you may discuss possible extensions of the work done or point out some connections between your subject and other applications or techniques, which you might have come across in the course of the work but which have not been been worked out in detail in the paper body.

The Introduction and Conclusion may contain other material that the author considers relevant, for example, a personal remark or opinion which cannot be conveniently expressed in the central, more formal part of the paper.

The size of the Conclusion should not exceed 1/2 of a page; in many cases, one or two paragraphs will suffice. Avoid trivial, non-informative phrases, like "Upon completion of this laboratory, certain conclusions can be drawn". An Introduction or Conclusion longer than one of the central sections is a sign that the material should be re-balanced.

2.3.6 Technical details

Other commonly used titles are “Method” or “Methodology”. Feel free to devise a subject-specific, explanatory title. If the original problem has several parts, subdivide this section accordingly. A subdivision may be needed simply for a better balance of section sizes or it can be demanded by the logic of exposition: if, say, different aspects of the method employ different techniques.

This section should describe all the fine (or heavy) details of the problem or model and the details of the mathematical method or algorithm used to solve it, as well as the structure and particulars of your computer code.

The requirements, in brief, are:

Attention to details and particulars. Relevance to the topic.

The subsection **Mathematical details** (or whatever your subject-specific title) generally includes notation used, definitions, mathematical formulation (set-up) of a model, relevant theoretical facts, and the mathematical essence of the algorithm used to solve the problem.

Do not attempt to reach far and wide; apply judgement. Suppose, for instance, that the problem is to find the distance between two skew lines in space. A student googles for *distance*, finds a Wiki article on *metric spaces* and blindly copies a definition to her paper. Big math, the prof must be pleased!? — Wrong! Irrelevant! (Plus, the level is inappropriate for the target audience, which is not the “prof”.)

Another student, faced with the equation $x^2 - 5x + 6 = 0$, engages in a lengthy step by step calculation using the quadratic formula. Five lines down, he finds the roots to be 2 and 3. It isn’t such a serious crime, but it wastes space on a triviality. In such simple cases, the answer ought to be given next to the equation. (In more involved cases, a detailed solution of a quadratic equation would make sense. Your reader may not immediately see that the roots of the equation $x^2 - 2tx + (t^2 - 1) = 0$ are $x_1 = t + 1$ and $x_2 = t - 1$.)

Neither speculating about things you don’t understand nor reiterating banalities is good. Focus on a content that’s not over your head and that really matters. Any symbol that appears in your calculations, arguments, or later in tables and graphs, should be defined. Definitions and terminology should be accommodated to the **concrete situation**. In an anecdotal case, a student writing a paper on graphs of polynomial functions started with a definition of a graph from Graph Theory!

Let us elaborate on **definitions** a bit further. They can be stated in two ways.

(1) As stand-alone structure units, in a separate paragraph, with the title word **Definition** in a distinguished font style. This format should be used when you introduce a major concept or when a definition is lengthy. (Do not hesitate to make definitions lengthy and detailed, even boring: precision and disambiguation are the priorities. Think of a legal code.)

(2) Inline definitions (as a part of the flow) can be used if the definition is very short, simple or natural, or if the concept is supposed to be generally familiar to the reader. Example: “To describe the shape of a rectangle with side lengths a and b, we introduce the parameter $\mu = b/a$ called the *aspect ratio*.”

The definitions, the notation, and the method should be described in such detail that a motivated reader could **reproduce** your work on his/her own and obtain identical results. How about yourself a few months later? Keep adding details and clarifying your writing until you are able to answer in the affirmative. We refer to Appendix B-1, Section 4, for further tips on mathematical writing. Let us just make one more suggestion regarding the mathematical method in general and contents of your *Mathematical details* section in particular.

Think about **simple particular cases**, where the situation is either obvious or the answer can be obtained easily. Do this before you write a program and before you explore the "real" data, for which you cannot predict the results. While creating your program, you will have convenient simple tests. For example, if you have to write a program to compute the area of a triangle with sides 14.23, 12.497 and 9.72, test your program on the Pythagorean triangle with sides $3, 4, 5$ first; also test it in the case where a triangle degenerates to a segment (say, for the sides 1, 2, and 3).

Also, think what happens when a certain parameter or a ratio of parameters becomes extremely large or extremely small. Quite often, intuition will suggest an answer and you'll be in a better position to make sense of the computed results.

Example 1. Suppose, as a part of the assignment, you have to construct a common tangent to two given circles. A circle collapses to a point when its radius tends to zero. Consequently, a common tangent to the two given circles becomes a line passing through the two given points when both radii shrink to zero. This limiting case provides a convenient test for your calculations and your computer program.

Example 2. Suppose the assignment asks you to find the number of grid points (whose both coordinates are integers) inside the circle of radius R centered at the origin. Think what happens as $R \to \infty$. Every grid point corresponds to the unit square of which it is the center, so the number of the grid points is approximately equal to the area of the circle, that is, πR^2. The fraction of the area contributed by incomplete unit squares overlapping with the circle is vanishingly small.

Considerations of this sort can make a valuable part of the *Mathematical details* section or of the *Results and Analysis* section.

The subsection **Program details** should provide a detailed breakdown of your program so that the reader can see how the mathematical ideas of the solution method are coded.

Please learn to differentiate between a mathematical method, or algorithm, and its programming implementation. Sometimes the description of the method and of the program, which implements it, can be intertwined, especially if the method is very straightforward. In other cases you are better to explain the method or its more subtle elements within *Mathematical details*, using conventional mathematical notation (dot or void for the multiplication sign, one-letter variables, subscripted if needed, etc.). The explanation of a program involves the actual syntax of the programming language used, with its own conventions ($*$ for multiplication, multi-letter names of variables, etc.). If necessary, explain the correspondence between mathematical variables and their counterparts in the program.

What is the best way to explain a computer program? On the one hand, from the end-user perspective the program is a black box that takes a specified input and produces an output, which the user should be in position to interpret. On the other hand, you must explain the internals, the workings of the code, and — primarily — the part pertaining to **mathematical operations**. It is the latter that we want you to emphasize in this section. You are writing a research report, not a user manual (a technical text, too, but of a different kind).

If your program validates initial data (reports an error on input of a negative distance, say) — good, but do not get overexcited about the user interface. It is better to make an effort to explain the overall logic of the program, flow control (loops, if/else operators), and the organization of data unless it is very obvious.

Example 1. Suppose your program counts the number of partitions $P(N)$ for N from 1 to 30. Your description of the program can begin as follows:

> Each run of the outermost loop of the program corresponds to computation of $P(N)$ for a particular value of N:
>
> ```
> DO N=1,MAXN
>   $\boxed{\text{Computation of } P(N)}$
> END DO
> ```
>
> The upper bound, `MAXN`, of the loop is set to 30 by default, but it can be changed through user's input.

Example 2. A line like this

```
DISTANCE=TIME*SPEED
```

is self-explanatory and doesn't need comments. A loop like this

```
DO I=TMIN+1,TMAX
 DISTANCE=DISTANCE+DT*SPEED(I)
END DO
```

can be commented on at the author's discretion, for example:

> In this loop, the distance traveled over time interval from `TMIN+1` to `TMAX` is computed. The array `SPEED` is initialized at the beginning of the program according to formula (2.3). [referring to the *Mathematical details* section] The variable `DT` in the program is time step, denoted by τ in the description of the method. Note that the value `SPEED(TMIN)` is not included, since it has been included in the previous summation.

The better your programming style, the easier it is to explain the program's overall design and logic. Look at Example 1 again. Perhaps, replacing the whole body of the loop (dozens of lines of code) by a framed summary was a good writing trick. But it becomes altogether unnecessary if a subroutine is used instead:

```
DO N=1,MAXN
   CALL NUM_PARTITIONS(N)
END DO
```

Self-explanatory names of variables, short functions, transparent if/else conditions, avoiding fancy syntax constructions like `cond[--i]=(i>=1)` in C, — all this helps.

Do not teach the reader the basics of programming. A general definition of a `for` loop copied from the Internet or a general discussion of the organization of computer memory or a definition of common data types (`int`, `double`) is not what is needed. If you really feel a need to remind the readers about some syntaxic details or other particulars of the programming language used — for instance, if you are writing for your own future reference — then, ok, find a place, but don't make it a big story. Even if you think (possibly correctly) that your instructor is not a Java or Python guru, it is not a reason to incorporate a section-long language tutorial.

When it comes to small details, give priority to those that require special care. Why does the value of `J` change from `0` to `N-1` and not from `1` to `N`? Why is the variable `numpoints` defined as `double`, while it naturally represents a positive integer quantity? (Conceivably, you want to allow it to assume very large values, beyond the range of the type `int`.) Issues like these can be addressed.

For the reader's convenience, include only short, critical fragments of the actual code in the body of the report. The complete listing can be included as an Appendix. In any case, the program code must be submitted electronically as a part of the assignment bundle.

The above is not dogma. For example, it can be important to explain to the reader how exactly your FORTRAN program produces the data file with coordinates — in which case the details of the `WRITE` operator should be discussed, although it is not a computational issue.

2.3.7 Results and Analysis

As in the case with the *Technical details* section, this one can be subdivided if the research has several parts.

In different projects, the meaning of "solution" or "results" is different. There have been few labs in this course where the answer can be stated in a really short form, like projects asking to decipher a coded message. In most labs, results come from a series of runs of a program that a student creates. Each single outcome can be just a number or it can be an array of numbers, a table or a graph.

As a bare minimum, your presentation of results should include:

- evidence that your mathematical method and your program are correct. Run sample cases that can be checked by hand calculation. Or demonstrate the workings of your program in cases where the solution is intuitively obvious.
- the solution(s) corresponding to those data provided in the assignment (if such data are indeed provided).

If correctness of the method/program cannot be demonstrated because the program doesn't work correctly, read Sect. 1.3.2.

Many projects in this course are to some degree open-ended. They ask you to go beyond the prescribed sets of data and to explore the problem further on your own. The assignment sheet may or may not give a hint on how to choose data for such experimentation. Some outcomes of your experiments will end up in a trash bin and some will make it into the paper. In the end, we want your Results to be more than just plural for a single 'result'. Interpret them, present them as a manifestation of a certain idea or phenomenon. We want you to spot a trend, to observe a pattern, to discover some sort of "law."

- Tell the reader why or how each example presented is relevant to the conjectured "law."
- Explain the observed pattern/law, at least partially.

The quality of your analysis of results and your mathematical explanations determine the research value of your paper more than anything else. Remember that this is a **mathematics** course and a large component of your grade will be based upon the paper's mathematical content. Don't just state observations. Analyse them and justify them! If the analysis and/or explanation of the results requires a piece of theory that has not been discussed in the *Mathematical details* section, include the necessary definitions and facts here, along with your own calculations and arguments.

In this section you can also explore the efficiency of your code: how fast or slow it is as the size of data fed to the program increases.

2.3.8 Acknowledgements and References

Any help that you have received from another person must be acknowledged. An acknowledgement should be expressed in the form of a grammatically complete sentence. If possible, specify the kind of help obtained. It also helps to characterize the status of the person so that those who come across your paper in a few years will know. Don't just say:

- *John Smith for his help with this assignment.*

Say instead:

> John Smith, a Computer Science major, has provided advice about the input/output functions in Java.

Or:

> I acknowledge help from Mr. John Smith, a tutor, in justifying the pattern as described in the Results section.

If you quote any printed material in your report, for example a calculus book, it must be listed as a reference. You should provide specific page number to enable the reader to easily find the place (definition, theorem, historical fact) you are referring to.

Example. In the body of the paper, put the reference number and the page number:

> If we substitute the elliptic arc equation (2) into the arc length formula [3, p. 548]
>
> $$L = \int_a^b \sqrt{1 + [f'(x)]^2}\, dx,$$
>
> we obtain the expression ...

In the References section, describe the source:

> [3] J. Stewart, *Calculus: Early Transcendentals*, 5th ed., Thomson-Brook/Cole, 2003.

(In this case, the publisher is a worldwide company and the place of publication is not indicated.)

There are different reference styles. Follow one style consistently. Check with your instructor as to whether a particular style is preferred.

Quoted online resources must also be given proper attribution. For instance, there is a webpage at `http://mathworld.wolfram.com/ContinuityPrinciple.html`. It has a title and, unlike, say, many pages in Wikipedia, it is not anonymous: we can name an author. For this particular page, a bibliographic entry would look like this:

> [1] Eric W. Weisstein, Continuity Principle, `http://mathworld.wolfram.com/ContinuityPrinciple.html`. (Accessed Dec. 5, 2008).

Alternatively, *"web sites may be cited in running text instead of in an in-text citation"* (The Chicago Manual of Style Online, section Website). This very line is an example.

The sections **Acknowledgments** and **References**, as well as **Abstract**, are not numbered. See Sect. 3.2 regarding LaTeX typesetting conventions for these sections.

2.3.9 Appendix

Material that does not naturally fit in the flow of your paper yet is important for your project's completeness should be put in an **Appendix**. Many papers will not have an appendix. Where an appendix is present, the following kinds of material are found in it:

- Computer code
- Graphs and illustrations
- Particularly long mathematical calculations or proofs

A listing of your program is the most common thing to put in the Appendix. Sect. 3.1.9 suggests how to get a nice printout.

A Math-2130 paper having more than one appendix should be an exception, but if that happens, identify the appendices alphabetically: **Appendix A**, **Appendix B**.

We urge you to learn quickly how to include graphs and illustrations **in the body of your report**. You may use `gnuplot`, `xfig`, the LaTeX picture environment, `Maple`, or PostScript. Graphs and illustrations which are not part of the text can comprise Appendix A of your report, while the computer programs can be presented in Appendix B.

2.4 Suggestions about style

2.4.1 A note on spelling

There are many spelling checkers available. Use them! On Linux, you can use `ispell`. The *`Kile`* editor has a built-in spell function which uses the `ispell` program. There is no excuse for spelling errors, and they will be **penalized** heavily. However, be warned, the spell checker will not find every error in spelling, nor can it pass judgement on a sentence like

> A program too gene rate asset off inter resting numb hers ...

Some common spelling errors

- their (whose?) — there (where?)
- its (whose?) — it's (it is)
- sep**a**rate (not sep**e**rate)
- occur**e**nce (not occur**a**nce)
- one's — once
- two — to — too
- then (if ..., then) — than (more than)
- lab$\boxed{\textbf{o}}$ratory (not labratory)

2.4.2 Squeeze water out (Eliminate unnecessary words)

Compare:
(1) It can be shown by the implementation of the Cosine theorem that the distance AB is equal to 5.
(2) Applying the Cosine theorem we see that $AB = 5$.

2.4.3 A note on “strong words”

Students often write: “it is necessary”, “one must” etc. Such strong expressions may justly raise objections.

Example. Suppose we are considering the equation $x^2 - 6x + 5 = 0$.

Bad description: *“It is necessary to use the Quadratic Formula. Thus we obtain...”*

Is it necessary? Absolutely not. Any one competent in quadratic equations will factor this one on the spot. If you want to emphasize the method, better say:

“The roots, as given by the Quadratic Formula, are $\frac{1}{2}(6 \pm \sqrt{D})$*, where* $D = 6^2 - 4 \times 5 = 16$*. Thus* $x_1 = (6 + 4)/2 = 5$*,* $x_2 = (6 - 4)/2 = 1$*. ”*

Better even (if the method is of no special importance) is just to say

“The roots are $x_1 = 5$ *and* $x_2 = 1$*”.*

Of course, the style and level of details that you should or should not provide depend on who your readers are. In any case, saying that to use the Quadratic Formula is *necessary* here is unprofessional.

2.4.4 Common words in mathematical writing

Learn to use basic mathematical terminology precisely and avoid common misuses. For example, watch the following as you write.

- An *equation* must have two parts (sides) connected by the = sign. A thing like $\sin^2 x + \cos^2 x$ is not an equation; it can be described as an *expression* or, more precisely, as a *trigonometric polynomial.* Also don’t call $x + y \geq 2\sqrt{xy}$ an equation; it is an *inequality.*

- At the beginning of a mathematical argument you often make an *assumption*, while at the end you arrive at a *conclusion.* Expressions like *Assume (suppose) that something is ...* or, equivalently, *Let something be* are very standard. Steps of your argument or formulas that you display or refer to should be verbally connected using words like *imply*, *follow*, *yield*, etc. to make the flow of the argument smooth and its logic transparent.

- Here are some common, frequently used, safe verb-noun collocations.

— An *equation* can be *solved* (or *solved for x*). In contrast, a *polynomial* (like $x^2 - 5x + 6$, without any right-hand sign) cannot be solved (there is no equation to solve) but it may *have roots.* Also, a polynomial can be *evaluated* at a particular point; equivalently, a particular *substitution* can be made into it.

— A *computer program* can *implement* a method or an algorithm; it is *created* or *written* by you; to obtain (produce) results you *run* (or *execute*) it; the results are *presented* or *displayed* in tables and graphs.

— A *value* can be *assigned to* a variable (or a variable can be assigned a value). Also, a value can be given *in a closed form* (as a mathematical expression that does not involve an approximation, e.g. $\sqrt{2}$ or $\arctan \pi/3$) or *approximately* (as a decimal fraction), e.g. 1.414 for $\sqrt{2}$.

— An *expression* (a *sum*, an *integral*, but not an equation) can be *evaluated* (calculated, computed); also it can be *transformed* (into an equivalent form), *expanded*, *factored*, *simplified*, *reduced* to a simpler form in a special (particular) case, *broken* into two or more parts (usually, to reveal an essential structure). But do not "solve a sum"!
— A new concept or notation can be *introduced*, *defined*, or described in a manner of this sort: *let us say that X is ...* [an explanation in terms of known or previously defined terms follows].

2.4.5 *We* versus *I*

Whether you should write in the 1st person (*we* or *I*) or in the 3rd person is to a large extent a matter of choice and personal style. It is more common to use *I* in the Introduction and Conclusion only, if at all. If you are bold enough to use *I* in the main part of your paper, do so consistently; do not hide behind *we* in "inconvenient" places.

That said, modern professional scientific writing overwhelmingly prefers *we*. Think about it this way: as a reader goes through your paper, he/she feels your company and gentle support: **"We = I, the author, + you, the reader"**.

You should use *I* when expressing personal opinions, e.g. "I found the results obtained in this laboratory truly surprising." There are other situations where the subject *I* sounds more appropriate than *we*. Consider this example: "I have encountered difficulty implementing this algorithm." Substituting *we* in place of *I* would imply that your innocent reader shares your responsibility for the trouble. Do not switch lightly between *we* and *I*. If the subject *I* is indeed necessary, its use should be confined to a particular section — better the Introduction or Conclusion — and there *I*, not *we*, should be used throughout.

While working on the project, you perform concrete actions, like searching a database, writing a program, etc. If you feel uncomfortable to write *we* when describing such actions, use sentences stated in the 3rd person. For example, "A program has been written" instead of "I have written a program". However, papers written in the 3rd person from the beginning to end usually leave an impression of an indirect, cumbersome style.

2.4.6 Verb forms: tense, mood, modal verbs

Use **present tense** predominantly when discussing mathematical theories and other abstract facts: they last indefinitely. Naturally, in talking about background and history of the problem you would use the past tense.

You may choose to use either past or present to describe how you have created a computer program and produced the results of numerical simulation. (The story may be the past for you when you submit the report, but it will be current for the reader as he/she reads it.)

Variation of tense can be employed just for a particular purpose. Do not constantly swing between one tense and another, especially within a paragraph or even a section.

Write in the **indicative mood** for the most part; use conditional only when it is unavoidable. Theorems *are* true, not *would be* true.

Watch other **modal verbs**, besides *would*. In many of students' papers the verbs *can* and *must* are unnecessarily frequent.

Example 1. Saying that "Equation (1) *can be shown* to yield formula (2)" leaves the reader wonder whether you omit significant steps. If the answer is yes, you ought to show *how* the former yields the latter. Otherwise (if the connection between (1) and (2) is transparent enough), simply say "Equation (1) yields formula (2)."

Example 2. Here *must* is used unnecessarily: "Therefore Equation (3) must hold when $t = 0$." It really means that "Equation (3) holds when $t = 0$." Say thus — shorter and better.