How to effortlessly write a high quality scientific paper in the field of computational engineering and sciences

Preprint · March 2020

DOI: 10.13140/RG.2.2.13467.62241

3 authors:

Vinh Phu Nguyen
Monash University (Australia)

Stéphane Pierre Alain Bordas
University of Luxembourg

Alban de Vaucorbeil
Deakin University

How to effortlessly write a high quality scientific paper in the field of computational engineering and sciences

Vinh Phu Nguyen[a,*], Stephane Bordas[b], Alban de Vaucorbeil[c]

[a] *Department of Civil Engineering, Monash University, Clayton 3800, VIC, Australia*
[b] *Institute of Computational Engineering, University of Luxembourg, Faculty of Sciences Communication and Technology, Luxembourg*
[c] *Institute for Frontier Materials, Deakin University, Geelong, VIC, 3216, Australia*

Abstract

Starting with a working good research idea, this paper outlines a scientific writing process that helps us to have a nearly complete paper when the last analysis task is finished. The key ideas of this process are: (1) writing should start early in the research project, (2) research and writing are carried out simultaneously, (3) best tools for writing should be used. The process seems working well as it has helped us writing thousands of pages without feeling a paint. We hope it works for you too. Due to our personal preferences, the discussion is confined to LaTeX based typesetting where we present guidelines to prepare high-quality images, tables, algorithms, source codes using LaTeX. We also discuss writing guidelines such as paper structure and some common mistakes.

Keywords: scientific writing; LaTeX; scientific publication; high quality writing.

1. Introduction

Publishing original research in a peer-reviewed journal is an important parameter to assess academic achievements. To get your paper published, not only its findings should be significant but it should also be well written such that these findings can be efficiently transmitted throughout the scientific community. The value of writing well should not be underestimated. Writing well leverages your work since 60% of reviewers criticisms pertain to the quality of the writing or tables and graphs with only about 40% to the quality of the scientific work (Iles, 1997). However, writing a high quality paper is not an easy task due to technical and language barriers. To help remove those barriers, books and articles giving advices on how to write scientific papers have been written (Day, 1998; Ashby, 2000; Plaxco, 2010).

Building on the advices found in the literature and our total 20 years of experiences, this brief paper gives guidance in writing high quality research papers. By high quality papers, we mean those of which scientific findings are not buried under a poorly written text. Different from existing related works, see e.g. Day (1998); Ashby (2000), we focus here on the tools or softwares that ease our writing process. Due to our background in computational mechanics – a sub-branch of computational engineering and sciences– we have used, as a model, a typical computational mechanics

*Corresponding Author
Email address: phu.nguyen@monash.edu (Vinh Phu Nguyen)

project: one develops a new model, implements it in a code, and carries out simulations using that code to demonstrate the performance of the model.

First, in Section 2, our favorite softwares are presented. These softwares are mostly open source and cross platform. Then, Section 3 provides actionable suggestions on how to structure your paper, avoid common mistakes. Particularly, we present an iterative writing process that is intertwined with your research. As we use LaTeX, a high-quality free typesetting system which is the *de facto* standard for the publication of scientific documents for hard sciences (see Table 1), we present some LaTeX related guidelines on how to prepare high-quality vector images, good-looking tables, algorithms, source codes and tweaks for two-column format papers (see Section 4). Finally, with the pressure to "publish" (or perish), it is increasingly difficult for students and researchers alike to resist the temptation to submit a "large" number of papers. Some of these might not be ready for submission yet. We present some considerations regarding this issue (Section 5).

Table 1: Summary statistics of the use of LaTeX in science disciplines (% of submitted papers) taken from `https://www.the-scientist.com/uncategorized/dont-format-manuscripts-44040`.

Disciplines	LaTeX rate
Mathematics	96.9%
Statistic and Probability	89.1%
Physics	74.0%
Computer Sciences	45.8%
Engineering	1.0%

There is no single, correct way to write. Our goal was not to convince you that what is described in this paper is the one true way, but instead to get people thinking, talking and sharing ideas about writing. Many researchers do not think much about writing because (many of) their papers get accepted and cited. What they do not know (or do not care about?) is that there are probably many readers struggling to grasp their papers. The situation might be worse as Judy Swan – Associate Director for Writing in Science and Engineering at Princeton Unviersity – saw it: 'scientific writing is bad writing'.

It is obvious that, no matter how good you are as a writer, a bad research idea will not result in a good paper. Therefore, we assume that you have had a sound research idea. This paper outlines a process that helps you to have a nearly complete paper when the last analysis task is finished.

The LaTeX source of this paper and various `Python` scripts used to prepare high quality images can be found at the github account of the first author: `https://github.com/vinhphunguyen/how-to-write-a-paper`.

2. Tools

It is obvious that using the right tool for any task is half way to success. And writing is no exception. Our favorite tools for writting papers are the following:

- LaTeX: a high-quality free typesetting system; it includes features designed for the production of technical and scientific documentation. It is the de facto standard for the communication and publication of scientific documents;
- `BibDesk`: an open-source reference management software package for macOS, used to manage bibliographies and references when writing essays and articles. It is primarily a BibTeX front-end for use with LaTeX (see Appendix C);
- `JabRef`: an open-source and cross-platform reference management software package. Its use is similar to `BibDesk`. It is primarily a BibTeX front-end for use with LaTeX. One great feature that we like: it fetches bibliography entries automatically when given a DOI;
- `Skim`: is an open-source PDF (Portable Document Format) reader for macOS. Alternatives are `Sumatra PDF` for Windows and `evince` for Linux;
- `Adobe Illustrator`: an industry-standard vector graphics software to create high quality drawings;
- `Inkscape`: a free and open-source vector graphics editor available for Linux, macOS, and Windows;
- `Matplotlib`: is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats. It can be used in Python scripts;
- `Dropbox`: on-going papers are stored in `Dropbox` so that we can access them from multiple devices;
- `Git`: is a distributed version-control system for tracking changes in source code during software development.
- `GitHub` or `Gitlab`: online platforms that offer hosting for software development version control using Git. They offer both public and private (preferred for research papers) repositories. When using Git, they make access from multiple devices and parallel collaboration easy.

The flowchart for the generation of a paper is as follows. LaTeX is used to typeset the paper. The references used in the paper are stored in a `.bib` file automatically generated and managed by `BibDesk` or `JabRef`. Sketches used in the paper are drawn using `Adobe Illustrator` or `Inkscape` and graphs are created using `Matplotlib`. Sketches and graphs are saved as pdf files and thus are of very high quality, see Section 4.2. Also, sketches and graphs could be generated such a way that the font used for embedded text is the same as that of your document.

To keep track of the changes in the code, figures, and even research data, `Git` is used. For collaborative writings, the git repository is hosted in either `Github` or `Gitlab`. They offer access to free private repositories that are very useful.

This is because every co-author of the paper can work on the same paper simultaneously. This paper was written that way. They can also use their own favorite LaTeX editor (which is most often also their own coding editor). We find this organization to be better than `Overleaf` at `https://www.overleaf.com`.

Remark 1. Without going in to the debate of which is the best editor, we use `Sublime Text`. This is because it can be used for both writing LaTeX documents and coding. Furthermore, it can render equations in real time, see Fig. 1, which is quite handy. And the two way sync between `Skim and Sublime Text` is super convenient. However, `Sublime Text` is proprietary. If you prefer open-sourced softwares, look into `Vim` and `Emacs`. Coupled with the `AucTex` package, `Emacs` can recognize LaTeX code and compile it easily. We prefer using a general purpose text editor rather than a LaTeX editor so that we just use one editor for both LaTeX typesetting and coding.

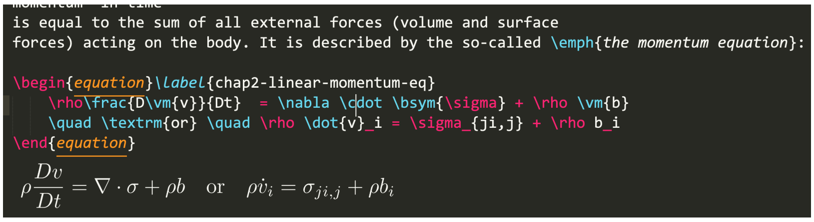


Figure 1: `Sublime Text` can render equations in real time.

Remark 2. There are many alternatives to `matplotlib such as gnuplot and Matlab`. We used `Matlab` and recently switched to `Python`. There are many reasons that influenced this, two of which are noteworthy. First, it is fun and beneficial to the brain to learn something new. Second, `Python` is now everywhere: it is used as a scripting language for many finite element packages (e.g. `Abaqus`) and scientific visualization programs (`Paraview` and `Ovito` for example).

The tools described in this section help you to write papers effectively, only when you have an idea how a paper should look like. The next section is devoted to just that.

3. Writing tips

This section presents writing tips. We start with a presentation of general guidlines in Section 3.1. Then we discuss the paper structure in Section 3.2. Some common mistakes and a few tips to make your paper less verbose or wordy are given next in Section 3.3. Finally, Section 3.4 – the most significant contribution of the paper – outlines an iterative writing process, losely inspired by the interesting talk by Jones (2016).

3.1. General guidelines

The following general guidelines for a high quality scientific paper are nothing new but they are worth being repeated:

1. To inform not to impress;
2. Aim for clarity and readability and reproducibility;

3. Contributions must be clearly stated;
4. Every unit of discourse (a sentence, a section, an article *etc.*), no matter the size conveys only a single idea or message (Gopen and Swan, 1990)[1];
5. Avoid jargon;
6. Minimize chances for reviewers to raise issues;

The main contributions of your paper must be clearly stated after a brief review of the literature: in which way your work differs from the existing literature. Be precise, as this is where the reviewers will try to find problems with your work. Their goal is to identify whether your work is novel or not. If it is not immediately clear from the Abstract and Introduction you risk being unconvincing.

Each paragraph conveys only a single idea or message. Do not be afraid of writing short paragraphs, even two-sentence ones. Use simple sentences that are linked together so that your writing is coherent. See Box 1 for a paragraph that was not well written: the second sentence is not related to the first one and 'this issue' in the third sentence was not clear. A better version is shown in Box 2 where the writing is more coherent: sentences start with familiar (old) information and end with unfamiliar (new) information. We prefer the new, important information at the end, because its job is to intrigue the reader. We refer to the old article of Gopen and Swan (1990) for more examples of writing readable paragraphs.

Try to revise your writing to keep only the words/figures/tables that are necessary. For example, using 'because' is more advisable than the wordy 'due to the fact that' (see Table 3 for a list of unnecessary words/phrases).

Box 1: Version 1.

In the phase-field modeling of fracture in brittle and quasi-brittle solids, it is crucial to represent the **asymmetric tensile/compressive material behavior**. **Existing phase-field models generally adopt either an intuitive split of the free energy density without capturing the crack boundary conditions properly or an ad hoc hybrid formulation at the loss of variational consistency**. **To address this issue**, this work presents a variationally consistent phase-field anisotropic damage model.

Box 2: Version 2.

In the phase-field modeling of fracture in brittle and quasi-brittle solids, it is crucial to represent the **asymmetric tensile/compressive material behavior**. **To capture this asymmetric behavior**, previous phase-field models generally adopt either and intuitive split of the free energy density without capturing the crack boundary conditions properly ... This work presents a phase-field anisotropic model that is able to capture **the asymmetric behavior**, **variationally consistent** and satisfy **crack boundary conditions**.

Avoid jargon which are the specialized vocabulary of any profession, trade, science. Writing a paper is not a race for complexity. You should make it as simple as possible for a neophyte reader to understand. Our advice is try to avoid jargon in the abstract and introduction as much as possible so that your paper is more accessible to a wide range

[1] Watch also the interesting talk of Judy Swan at `https://www.youtube.com/watch?v=1pzjxYCwb08`.

of audience. In case that a jargon is needed, provide a definition for it the first time it appears in the paper, and also include clarifications for any poorly formed jargon.

```
There are different formulations  to model the tension/compression asymmetry of fracture.
```

The jargon 'tension/compression asymmetry' might not be clear to some audiences. A better version of the above is:

```
There are different formulations  to model the tension/compression asymmetry of
fracture - fracture does not occur in domains under compression.
```

Reproducibility is a big issue in scientific research nowadays. However, we just confine ourselves here to the situation where a published simulation result is genuinely correct but impossible to reproduce by people other than the authors. The world would be a better place if all authors are more thoughtful when reporting their results: all information needed to make that particular simulation work should be provided. Particularly, nontrivial parameters.

You can save time for both you – the authors – and the reviewers by not making them guess. For example, if you do not do large deformation simulations, make it clear and justify that choice. If you have used a particular value for one numerical parameter, explain your choice. If reviewers have to guess your choices, they will comment on that. This increases the chances for your paper to be rejected, or needing corrections.

3.2. How to structure your paper

An excellent article on how to structure a scientific paper can be found at `https://www.nature.com/scitable/topicpage/scientific-papers-13815490/`. However, they are just general guidelines. A good strategy for getting started is to study how others have structured their papers. Select a couple of papers that you enjoy and understand and study how they are organized. You will learn a lot from copying these papers. Our favorite author is Ted Belytschko.

We will not repeat the Nature paper guidelines herein. Instead, we elaborate on some of the arguments such as a complex section should have a global paragraph before going into its subsection (Section 3.2.1), how to write a compelling introduction section (Section 3.2.2), and what the conclusion section should include (Section 3.2.3). Appendices and footnotes are discussed in Section 3.2.4 and references in Section 3.2.5.

3.2.1. Global paragraph for long sections

For paragraphs that are quite complex it is a good idea to write a global paragraph (or mini-introduction) between the heading of a section and the heading of its first subsection. Remember to prepare your readers for the structure ahead at all levels. See Fig. 2 for an example.

Papers on the field of computational engineering and sciences always have a section, typically named 'Numerical Examples' where some tests are presented to demonstrate the performance of the model/theory presented. While these

4. Homogeneous solutions

This section presents the analytical homogeneous solution for a bar under uniaxial tension. For simplicity, no body force and acceleration are considered. Furthermore, only monotonic loadings are assumed. This exercise serves multiple purposes: (i) it helps us understand the model, (ii) it demonstrates that some models are sensitive to b, (iii) it is a good test to check the FE implementation.

We plan this section as follows. The homogeneous solutions for the damage and stress field are treated in Section 4.1. Then, comparative studies of Neo-Hookean-I and Neo-Hookean-II and of `AT1/2` and `PF-CZM` are presented in Section 4.2.

4.1. Homogeneous damage and stress field

Figure 2: A complex section should have a global paragraph between the heading of a section and the heading of its first subsection.

examples are most often presented in order of increasing complexity, we can do a better job in presenting them. For example, a global paragraph stating why these examples were chosen, which open source (if it is the case) code was used, etc. A table with all parameters used for all simulations would be helpful, see Table 4.

3.2.2. Introduction section

Everyone would agree that the introductory section of a paper should contain the following items, in order:

1. What the problem that the paper is solving;
2. Demonstration the importance of that problem;
3. What are the current approaches to solving this problem and what is wrong about them;
4. What are the contributions of the paper;
5. Planning the readers for reading the subsequent sections.

Writing an introductory section that simultaneously (i) includes all the above items, (ii) covers all relevant works, (iii) is easy to follow and (iv) is short is not an easy task.

What we commonly see is introductory sections of about 2 to 3 pages, full of just plain text with lots of jargons. There are two problems with this type of writing. First, only the authors and a dozens of experts can understand what is going on. Second, the paper loses many readers. We have realized that using some formula, figures, tables in the introduction section significantly improves the readability. See Box 3 for an example, taken from Mandal et al. (2019).

Box 3: Equations and tables can improve the introduction section.

According to second-order PFMs for quasi-static fracture of solids under the infinitesimal strain regime, the displacement field $\boldsymbol{u}$ and damage field d are minimisers of the following total energy functional of the solid

$$\mathscr{E}(\boldsymbol{u}, d) = \int_{\Omega_0} \left[\omega(d)\psi_0^+(\boldsymbol{\epsilon}(\boldsymbol{u})) + \psi_0^-(\boldsymbol{\epsilon}(\boldsymbol{u}))\right] \mathrm{d}V + \int_{\Omega_0} \frac{G_\mathrm{f}}{c_\alpha} \left[\frac{1}{b}\alpha(d) + b\,(\nabla d \cdot \nabla d)\right] \mathrm{d}V - \mathscr{P}(\boldsymbol{u})$$

where the first integral is the stored strain energy, the second one denotes the fracture energy à la Griffith. The positive and negative parts of the strain energy density are denoted by $\psi_0^+(\boldsymbol{\epsilon}(\boldsymbol{u}))$ and $\psi_0^-(\boldsymbol{\epsilon}(\boldsymbol{u}))$, respectively.

model	$\alpha(d)$	$\omega(d)$	fracture type	length-scale	sup.	Parameters
`AT2`	d^2	$(1-d)^2$	brittle	$b = (27/256)l_\mathrm{ch}$	∞	$E_0, \nu_0, G_\mathrm{f}, b$
`AT1`	d	$(1-d)^2$	brittle	$b = (3/8)l_\mathrm{ch}$	$4b$	$E_0, \nu_0, G_\mathrm{f}, b$
`PF-CZM`	$2d-d^2$	$\dfrac{(1-d)^p}{(1-d)^p + Q(d)}$	brittle/cohesive	numerical param.	πb	$E_0, \nu_0, G_\mathrm{f}, f_t$

One way to visually demonstrate the contributions of your paper is to use a table in which a comparison with existing models is given. We borrowed this idea from the computer graphics community, see e.g. Stomakhin et al. (2013). Such a table is Table 2 where we compared different variants of the material point method (MPM).

MPM variant	Efficiency	Quad. error	Cell crossing	Num. fracture	Grid type	Contacts
MPM	☺ ☺ ☺	☹ ☹ ☹	yes	yes	Cartesian/unstructured	☺
GIMP	☺ ☺	☹ ☹	no	yes	Cartesian	☺
CPDI	☺ ☺	☹	no	no	Cartesian	☺
TLMPM	☺ ☺ ☺ ☺	☹	no	no	Cartesian/unstructured	☹
iMPM	☺	☹	no	n/a	n/a	n/a

Table 2: Overall characteristics of common MPM variants. The smileys and frownies are typeset using the package `wasysym`.

Another way to intrigue readers is to summarize all the impressive simulations that the model presented in your paper can do in a figure (see Fig. 3). In this way, you increase the chance that your paper gets read to the end.

3.2.3. Conclusion section

There are two misunderstandings about the Conclusion section. First, the Conclusion section is usually made long under the false belief that a longer Conclusion will seem more impressive. Second, it is most often just a replication of the Abstract and/or part of the Introduction in a present perfect tense. If the reader has to read your Conclusion to know what your paper is all about, then your Abstract and Introduction were not well written.

Some papers with well written sections even do not have a conclusion section. We are not a fan of that and believe having a conclusion is a good thing. Following the Nature paper introduced at the beginning of this section, a well

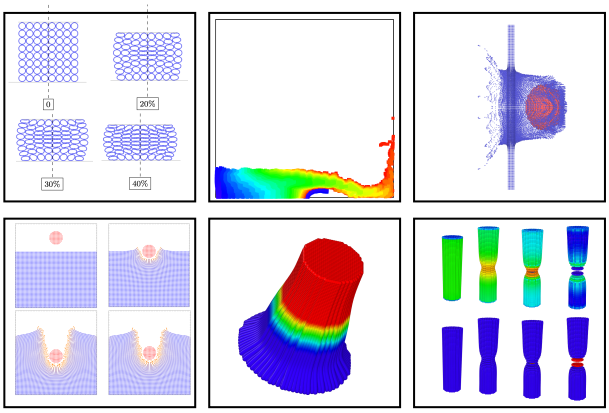


Figure 3: A figure with all good results that the model in your paper can deliver can help to intrigue readers so that they will read your paper to the end. Taken from de Vaucorbeil et al. (2019).

written conclusion section should include the following items:

- One or two sentences summarizing what the paper has been about;
- Summarizing the key findings of the paper;
- What could be improved.

3.2.4. *Appendix and footnotes*

It is appropriate to include appendices when the incorporation of material in the body of the paper would make it poorly structured or it would be too long and detailed and to ensure inclusion of supporting material that would otherwise clutter or break up the narrative flow of the paper. As discussed later in Section 4.4, it is better to move some equations to an appendix.

There are opposite ideas about footnotes. Some authors use them scarcely and some use them extensively. The argument of the former is that footnotes *break the flow of thoughts and send your eyes darting back and forth while your hands are turning pages or clicking on links* according to Pulitzer prizewinner novelist Cormac McCarthy (see `https://www.nature.com/articles/d41586-019-02918-5`). The argument of the latter is probably to reduce the paper's length. But, we find one wrong thing about footnotes: too lengthy footnotes. Some papers contain footnotes that are half a page long and with a smaller font than the main text. These are not readable.

We use footnotes sparingly and they are most often short. If you find a long footnote, consider using a remark as we have done in Remark 2.

3.2.5. *References*

You should pick one reference style and stick to it so that the references are consistent. Regarding how relevant work should be cited, below are some suggestions:

- Cite originals not derivatives;
- Avoid citing a list of two many papers *e.g.* 'See [1-20] for some relevant work'. This helps neither the readers to find anything, nor the authors of [1-20] to get credit;
- If a author-year reference format is used *e.g.*'Walker (1996) studied ...', all references in a single citation should be ordered in chronological order. For instance, 'To help ease the writing process, books and articles giving advice on how to write scientific papers have been written (Day, 1998; Ashby, 2000; Plaxco, 2010).'

3.3. Some common mistakes

Some common mistakes are given in Table 3. You can avoid these mistakes by studying the writing style of your favorite author. Don't worry too much about grammatically perfect sentences. It is more important to be understood.

Don't/Avoid	Do/Use
The Table/Figure 2	Table/Figure 2
The Equation (2.2)	Equation (2.2)
The Young's modulus	Young's modulus, or the Young modulus
Start a section with a table/figure/equation	Start a section with text
This topic has interested researchers for a **long** time	... for more than 20 years
A **bad** result	A poor/negative result
This section **serves** to explain	This section explains
It is **obvious/clear** ...	
Due to the fact that ...	Because ...
It should be noted that there are 5 samples in this study	This study consisted of 5 ...
In order to include ...	To include ...
The difference was **found to be** significant	The difference was significant
We plotted the data **by** using ...	We plotted the data using ...
Utilize or usage	Use
We **think/believe/feel** that the results are good	The results are good
Existing works ...	Previous work
Using adjectives such as 'very', 'always', 'never'	
Using words like 'ground-breaking', 'paradigm shift'	
Using 'Above-mentioned' or 'aforementioned'	
	Always spell out an acronym the first time it is used
Use long titles	Use short titles (Paiva et al., 2012)
	Use a spell checker to get rid of all spell errors

Table 3: Some common mistakes.

Sentences can be described as active or passive. Using the passive voice is a way of writing sentences so that the subject has the action done to it. A common belief is that the passive voice can be useful for making writing sound more formal and objective. However, using it extensively results in papers which are boring with hard to understand lengthy sentences.

On the other hand, using a personal tone can help to engage a reader. And the sentences are shorter and thus easier to understand.

3.4. *Writing process: an iterative process*

The first idea is that when you have finished your last simulation, the first draft of your full paper is complete. Here, by *you*, we mean the co-author of the paper who is in charge of the writing. After that, it just comes down to polishing the paper. The second idea is to not lose motivation due to set backs. That is, if the simulations are not working, don't be upset; let's write something instead. It can be as easy as filling Section *Acknowledgments*. Having updated your paper will definitely make you feel good. And that is very important. The third idea is that writing is intertwined with all other activities (formulation, coding and running simulations) as illustrated in Fig. 4.

After a research idea has been developed, you should start writing the paper (Gray, 2005). Obviously, the paper is empty, see Listing 1 for a TeX file for an empty paper. For the sake of presentation, let's assume that you need to develop a formulation, implement it in a code, and carry out simulations using this code. You first work on the formulation. Then, when there is some progresses, you can write some key equations in the paper (filling Section *Methodology*). Having the formulations nicely written in a PDF can help you to spot errors and to crystallize your thinking. Now that the formulation is complete, let's move to the implementation. Again, this task should be intertwined with the writing as well (filling Section *Methodology*). Most often, you start with a very simple problem to test the code (and the idea). If this example works, you are confident about the idea, you can write something on Section *Introduction* while the second simulation is under way. If this second example is important, you can write about it in Section *Examples*. If you are lucky, the result of this second simulation is good. Bingo, you can now fill Sections *Introduction* and *Abstract* while working on the third simulation.

Listing 1: A starting TeX file. This is for papers to be submitted to Elsevier journals. Change the template (documentclass) accordingly for other publishers. Introducing a table of content helps to see the overall structure of the paper. For brevity, packages used were skipped, see Listing 2. TeX keywords are highlighted in **bold blue font**.

```
\documentclass[authoryear,3p,times,preprint,review,fleqn]{elsarticle}
\title{\textbf{}}
\begin{abstract}
\end{abstract}
\tableofcontents          % TOC
\section{Introduction}
\section{Methodology}
\section{Examples}
\section{Conclusions}
\section*{Acknowledgments}
\bibliographystyle{abbrvnat} % bib style
\bibliography{mpm}           % bib file
```

If you feel stuck at writing any parts of the paper, feel free to do something else because keeping focusing on the writing does not always help. Most often, ideas come when you are in a diffuse mode, a concept proposed in Oakley et al. (2018). For example, while playing with your kids on a playground, the idea for writing a good abstract usually

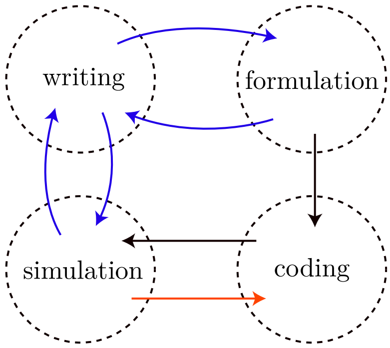


Figure 4: Writing is intertwined with other activities of the project.

comes. Jotting down the idea on a phone and you're done with this part of the paper.

While working on this paper, you read the literature (we always read it anyway). If you find a good paper relevant to your work, put it in `Bibdesk or JabRef`, and cite it in the paper with some key sentences about it. Doing so saves you a lot of time by not re-discovering this paper in the future. Note that `Bibdesk` can link a PDF to a paper. Therefore, we can have a library of papers on top of a `.bib` file.

Continuing this process, by the time the final simulation has been finished, you already have a nearly complete paper. Note that you have already revised your paper many times when your simulations were running (which usually take a long time). You just need to write the conclusions. And voila, you have a complete paper. Before submission, there are some steps discussed in Section 5 that need to be done.

Don't worry about the size of the paper while you are working on it. Put as many details as you feel needed. You might end up with, not a paper, but a long report (but in a format of a paper). If this is the case, keep this report (which can be used later, for instance, in your books), save it as another TEX file and remove unnecessary parts.

Now you know how a good quality scientific paper looks like and you are ready to compile such one. The next section presents some suggestions on how to do this electronically using LATEX.

4. LATEX tips

We present in this section some LATEX tips which have been collected over the years. In Section 4.1, we list the must-have packages. Then, we discuss how to prepare high quality plots in Section 4.2, tables in Section 4.3, notations and equations in Section 4.4, algorithms in Section 4.5 and source code in Section 4.6. Modifications required for preparing two-column format papers are presented in Section 4.7.

4.1. Packages

To improve the writing experience, once in a while one should update their LATEX skills. Listing 2 provides an updated list of LATEX packages being used to write our papers. We do not plan to discuss all the packages. Instead, we want to discuss two packages: `hyperref` and `cleveref` which are useful. By setting the option *backref=page* for the package `hyperref`, there appears ‘Cited on page #’ at the end of all references which are back-references to the page(s) in which a given reference was cited.

Using standard cross-referencing in LaTeX only produces the label number, a name describing the label such as figure, chapter or equation has to be added manually. The `cleveref` package overcomes this limitation by automatically producing the label name and number:

```
\cref{fig:figure1},  instead of Fig.~\ref{fig:figure1}
\cref{eq:equation1}, instead of Eq.~\ref{eq:equation1}
```

Listing 2: Commonly used LaTeX packages.

```
\usepackage{amsmath,amssymb, mathtools,mathrsfs,stmaryrd,titletoc}
\usepackage{natbib}
\usepackage[scaled=0.92]{helvet}  % set Helvetica as the sans-serif font
\renewcommand{\rmdefault}{ptm}    % set Times as the default text font
\usepackage[retainorgcmds]{IEEEtrantools}
\usepackage[usenames]{color}
\usepackage{tabularx}    % tables
\usepackage{booktabs}    % better tables
\usepackage{multirow}    % multi-row tables
\usepackage[font=small,labelfont=md]{caption,subfig} % sub-figures (see Fig. 6)
\usepackage[T1]{fontenc} % typing french
\usepackage[bookmarks=true,colorlinks=true,linkcolor=blue,backref=page]{hyperref}
\usepackage{float}          % make new float environment such as boxes (captioned)
\usepackage{listings}       % insert source code, used herein to insert LaTeX and Python codes
\usepackage{algorithm}      % flow wrapper for algorithm,similar to block commands table/figure
\usepackage{algpseudocode} % second algorithm typesetting environment (actual pseudocode)
%microtype: for Micro-Typographic Improvements
\usepackage[activate={true,nocompatibility},final,tracking=true,kerning=true,
spacing=true,factor=1100,stretch=10,shrink=10]{microtype}
\usepackage{nicefrac}               % type inline fractions: \nicefrac{1}{2}
\usepackage{numprint}               % \numprint{10000} => 10 000 not 10000
\usepackage[title,titletoc,toc]{appendix}
\usepackage[capitalise]{cleveref} %Basically, cleveref must be loaded last.
\definecolor{darkgray}{rgb}{0.95,0.95,0.95} % color used in tables
% cleverref package: just do \cref{label} for figures, tables, equations anything
% the package will determine the correct prefix be it Fig. or Equation or Listing.
\crefname{figure}{Fig.}{Figs.}
\crefname{equation}{Equation}{Equations}

\renewcommand*{\backref}[1]{}
\renewcommand*{\backrefalt}[4]{[{%
    \ifcase #1 %
          \or Cited on page~#2%
          \else Cited on pages #2%
    \fi%
    }]}
```

4.2. Figures

Figures are an important element of reporting the findings of your research. A high quality figure is worth a thousand words. By a high quality figure we meant one that has a legible font size, a high resolution, color-blindness aware, all the axes are clearly defined, those sorts of things. This section discusses how to generate high quality figures: be it graphs (Section 4.2.1), sketches (Section 4.2.2) and contour plots (Section 4.2.3). It also discusses on how figures/tables should be referred to in the paper (Section 4.2.4).

4.2.1. Graphs

It is not a requirement that the font used in graphs (*e.g.* bar charts, error chats, x-y scatter plots *etc.*) match that of the text. Yet, it would be better if they match. Using `Matplotlib`, one can generate graphs scatterplots which either are PDFs with font nearly matching the text font (see Fig. 5) or PGFs (Portal Graphic Format) with matching font (see Fig. 6). The LaTeX code used to include this type of plot is shown in Listing 3. We refer to Listings 9 and 10 in Appendix A for the `Python` source codes that generate these PDFs and PGFs from a data file.

Listing 3: LaTeX commands to insert either a PDF, or PGF or PDF_TEX image. The crucial point here is not to scale the inserted image. Otherwise, the font size will be affected.

```
\begin{figure}[!ht]
  \centering
  % only one of the following three
  \includegraphics{cold-spray-plots.pdf} % insert a PDF
  \input{cold-spray-plots.pgf}           % insert a PGF
  \input{output.pdf_tex}                 % insert a pdf_tex
  \caption{Cold spraying with a single impact: evolution of plastic strain and temperature.}
  \label{fig:cold-spray}
\end{figure}
```

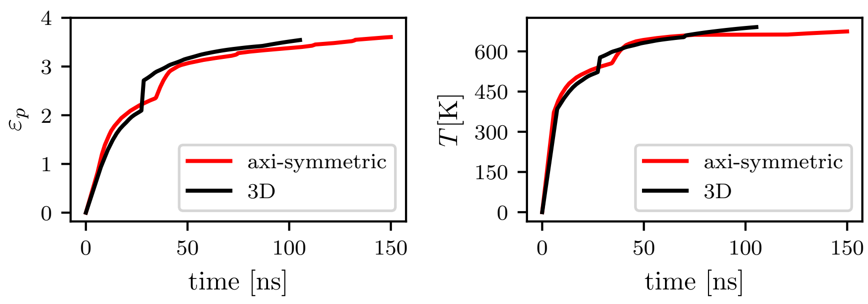


Figure 5: A PDF figure of which the font nearly matches the text font: evolution of plastic strain ε_p and temperature T in time. Symbols, if any, in figures should be typeset with LaTeX. Be thourghful about colour blindness that affects around 8% of men, particularly an inability to distinguish red and green. `matplotlib` can be colour-blind appropriate, see line 29 of Listing 9. Also, graphs should not have a title. Put the title in the figure caption.

If you want to stack multiple pictures together with sub-captions using LaTeX, the package `subfig` can do the job. Fig. 7 is a collection of 4 figures, with caption for each one of them. The corresponding LaTeX code is given in Listing 4.

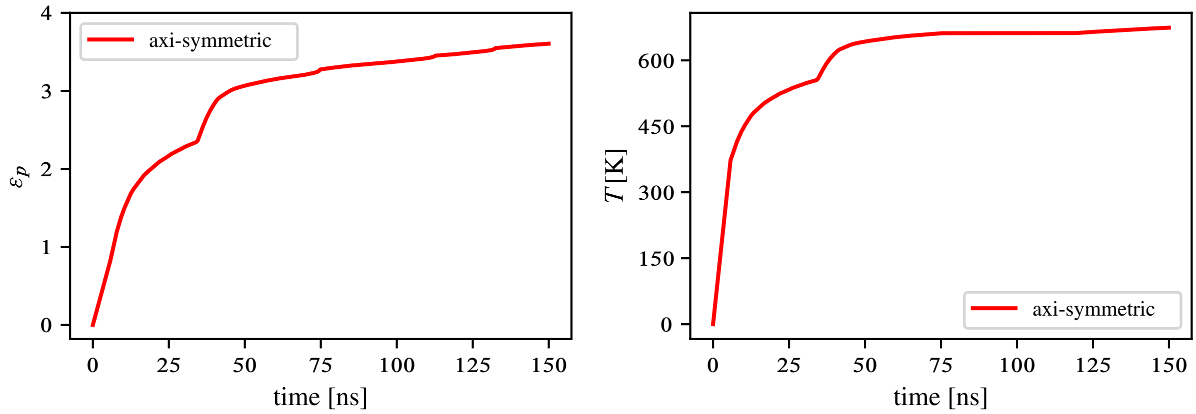


Figure 6: A PGF figure of which the font matches the text font: evolution of plastic strain ε_p and temperature T in time. PGF pictures are embedded as raw commands in LaTeX documents and thus can slow down the compilation process (*i.e.*, the process from a TeX file to the final PDF).

4.2.2. *Sketches*

If you use `Illustrator` for some drawings and need to include mathematical symbols in them, then try the application called `LaTeXiT`[2] to typeset whatever symbols and drag and drop them to `Illustrator` as embedded PDFs[3]. Fig. 8a presents an example. The same thing can be done using `Inkscape` with some LaTeX extensions. This is done by writing all text or formula using LaTeX syntax in `Inkscape`. Saving the drawing into svg, and then exporting it as a PDF_TEX using the following command (Engelen (2010)):

```
inkscape -z -D --file=input.svg --export-pdf="output.pdf" --export-latex
```

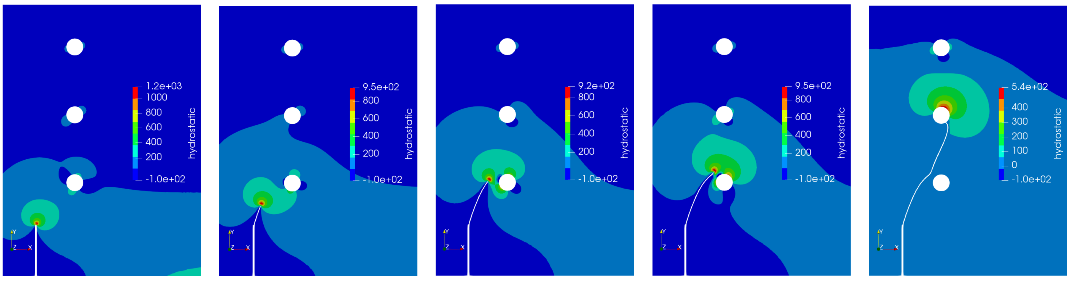
Finally inserting the figure is done according to Listing 3 (use line 5). Fig. 8b presents an example.

Some people even go to the extreme of not using a graphics software with a user interface (e.g. `Illustrator`). Instead, they use `TikZ`, a TeX package for creating graphics programmatically. It has a steep learning curve but the results are outstanding. An example is provided in Fig. 9. The code used to generate this drawing is given in Listing 5. One can appreciate the fact that it is fully parameterized, *i.e.*, the size of the drawing is controlled by the variable L.

4.2.3. *Contour plots*

Another type of images in scientific papers is contour plots (see Fig. 7), which are the outcomes of some visualization applications such as `ParaView`. These images should contain a color bar, and color bars limits should be the same on contour plots you are comparing, otherwise it is very difficult to compare. We do not know how to match the font used in these images with the one in the text. We simply save them as PNG files of biggest resolution and include them in the TEX document in the same manner as PDF images.

[2]This app can be found at `https://www.chachatelier.fr/latexit/`.
[3]Presentations can be created using `Keynotes` with equations created using `LaTeXiT` in exactly the same way.

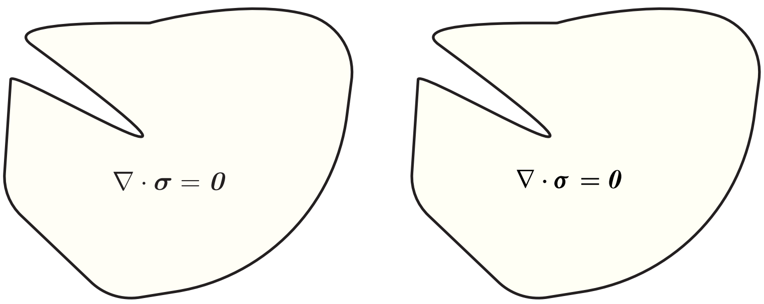


(a) CMOD = 0.008 inch (b) CMOD = 0.010 inch (c) CMOD = 0.012 inch (d) CMOD = 0.016 inch (e) CMOD = 0.018 inch

Figure 7: Stacking multiple pictures together with sub-captions using the package `subfig`.

Listing 4: Stacking multiple images using LaTeX package `subfig`.

```
\begin{figure}[h!] \centering
\subfloat[CMOD=0.008 inch]{\includegraphics[width=0.18\textwidth]{t14} \label{fig:a}}\;
\subfloat[CMOD=0.010 inch]{\includegraphics[width=0.18\textwidth]{t16} \label{fig:b}}\;
\subfloat[CMOD=0.012 inch]{\includegraphics[width=0.18\textwidth]{t20} \label{fig:c}}\;
\subfloat[CMOD=0.016 inch]{\includegraphics[width=0.18\textwidth]{t28} \label{fig:d}}\;
\subfloat[CMOD=0.018 inch]{\includegraphics[width=0.18\textwidth]{t30} \label{fig:e}}\;
\caption{Asymmetric notched beam under three-point bending: crack evolution.}
\label{fig:bittencourt-evolution}
\end{figure}
```



(a) `Illustrator` and `LaTeXiT` (b) `Inkscape` and PDF_TEX

Figure 8: Using `Illustrator` and `Inskcape` to produce vector images with LaTeX symbols. The font in figure (a) is slightly different from the one in the text ($\nabla \cdot \boldsymbol{\sigma} = \boldsymbol{0}$), while it matches perfectly in figure (b).

Pay attention to the figure captions. Ideally, the reader should be able to ascertain the entire story just by reading the figure captions *i.e.*, without going back and forth between the figure and the text sections.

4.2.4. Citing figures

The number one practice about citing figures is remember to cite all the figures. Even though there are some LaTeX packages *e.g.* `Refcheck` that help to find out unused figures and tables, our practice is when you insert a figure in your document, cite it immediately.

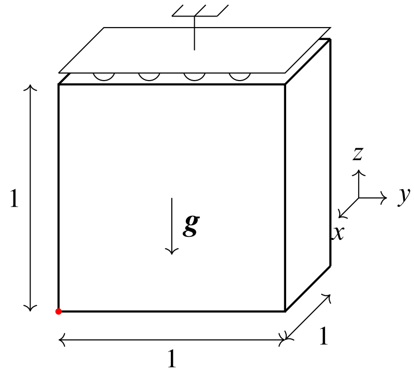


Figure 9: Example of schematic made using `TikZ`.

One usually does not pay enough attention to how figures (and tables) should be referred to in the text. That is why one usually write the following sentences

```
Fig. 10 depicts the global responses. As can be seen, the global responses are
insensitive to the incorporated length scale.
```

The first sentence is unnecessary as it only directs the reader to the figure (Fig. 10), and thus it provides no information. We have found similar sentences in the literature. A better version of the above is:

```
The global responses are insensitive to the incorporated length scale (Fig. 10).
```

4.3. Tables

Tables in scientific papers should be clear and focus on the data. Here are some suggestions for making good tables: avoid vertical lines, avoid double horizontal lines, avoid boxing up cells and leave enough space between rows. Table 4 satisfies all the criteria. The LaTeX code is shown in Listing 6. Table 7 in Appendix B presents a table with footnotes.

4.4. Notations and equations

Avoid overly complex notations, see Table 5 for some examples. Try to introduce a nomenclature in the paper, to ease things for the reviewers and readers. Sometimes, a table such as Table 6 does a good job to introduce main notations.

Ideally, the impact of a scientific work should be determined by its scientific merit, rather than by presentational style. Unfortunately, Fawcett and Higginson (2012); Higginson and Fawcett (2016) showed that scientifically strong papers may have reduced impact if not presented in an accessible manner. The density of equations in an article on ecology and evolutionary biology has a significant negative impact on citation rates, with papers receiving 28% fewer

Listing 5: TikZ code used to obtain Fig. 9.

```
\begin{tikzpicture}[darkstyle/.style={circle,draw,fill=gray!40,minimum size=20}]
  \tikzmath{
    \L = 3;
    \h = \L / 30;
  }

  \draw [thick] (-\L/2,-\L/2) rectangle (\L/2,\L/2);
  \draw [thick] (\L/2, -\L/2) -- (\L/2+\L/5, -\L/2+\L/5);
  \draw [thick] (\L/2+\L/5, -\L/2+\L/5) -- (\L/2+\L/5, \L/2+\L/5);
  \draw [thick] (\L/2, \L/2) -- (\L/2+\L/5, \L/2+\L/5);
  \draw [thick] (-\L/2+\L/5, \L/2+\L/5) -- (\L/2+\L/5, \L/2+\L/5);
  \draw [thick] (-\L/2, \L/2) -- (-\L/2+\L/5, \L/2+\L/5);

  \draw [->] (0,0) -- (0,-\L/4);
  \node [right,thick] at (0, -\L/8) {\large $\boldsymbol{g}$};

  \draw [<->] (-\L/2,-\L/2 - \L/8) -- (\L/2,-\L/2 - \L/8);
  \draw [<->] (\L/2,-\L/2 - \L/8) -- (\L/2+\L/5,-\L/2 - \L/8+\L/5);
  \draw [<->] (-\L/2 -\L/8, - \L/2) -- (-\L/2 -\L/8, \L/2);
  \node [anchor=east] at (-\L/2 - \L/8,0) {1};
  \node [anchor=north] at (0,-\L/2 - \L/8) {1};
  \node [anchor=north west] at (\L/2+\L/10,-\L/2 - \L/8+\L/10) {1};
  \draw[] (-\L/10, \L/2 + \L/15) circle (\L/20);
  \draw[] (-\L/5-\L/10, \L/2 + \L/15) circle (\L/20);
  \draw[] (\L/10, \L/2 + \L/15) circle (\L/20);
  \draw[] (\L/5+\L/10, \L/2 + \L/15) circle (\L/20);
  \draw [-, fill=white] (-\L/2,\L/2 + \L/20) -- (\L/2,\L/2 + \L/20) --%
  (\L/2+\L/5,\L/2 + \L/20+\L/5) -- (-\L/2+\L/5,\L/2 + \L/20+\L/5) -- (-\L/2,\L/2 + \L/20);
  \draw [-] (\L/10,\L/2 + \L/20+\L/10) -- (\L/10,\L/2 + 3*\L/10);
  \draw [-] (\L/10-\L/10,\L/2 + 3*\L/10) -- (\L/10+\L/10,\L/2 + 3*\L/10);
  \draw [-] (\L/10-\L/10,\L/2 + 3*\L/10) -- (\L/10-\L/10 + \L/20,\L/2 + 3*\L/10 + \L/20);
  \draw [-] (\L/10,\L/2 + 3*\L/10) -- (\L/10 + \L/20,\L/2 + 3*\L/10 + \L/20);
  \draw [-] (\L/10+\L/10,\L/2 + 3*\L/10) -- (\L/10+\L/10 + \L/20,\L/2 + 3*\L/10 + \L/20);

 % Triad:
  \draw [->] (\L/2+\L/5+\L/8,0) -- (\L/2+\L/5+2*\L/8,0);
  \draw [->] (\L/2+\L/5+\L/8,0) -- (\L/2+\L/5+\L/8*0.29,0-0.71*\L/8);
  \draw [->] (\L/2+\L/5+\L/8,0) -- (\L/2+\L/5+\L/8,0+\L/8);
  \node [anchor=north] at (\L/2+\L/5+\L/8*0.29,0-0.71*\L/8) {$x$};
  \node [anchor=south] at (\L/2+\L/5+\L/8,0+\L/8) {$z$};
  \node [anchor=west] at (\L/2+\L/5+2*\L/8,0) {$y$};
  \draw [fill=red, red] (-\L/2,-\L/2) circle (1pt);
\end{tikzpicture}
```

citations overall for each additional equation per page in the main text (Fawcett and Higginson, 2012). For papers in physics, the number is 6% fewer citations for each additional equation per page (Higginson and Fawcett, 2016).

The lesson to learn from the above relation between the density of equations in a paper and its impact is to write less equations in the body of the paper. This can be achieved by removing unnecessary equations. If needed, some equations can be put in appendices.

Table 4: Material parameters and characteristics for all simulations.

Parameter	Section 5.1	Section 5.2
Young's modulus [MPa]	210×10^3	145
Poisson's ratio [-]	0.3	0.45
Tensile strength [MPa]	2445	20
Experimentally validated	n/a	n/a
Solver	multi-step AM	single-step AM implicit-explicit
State	Plane strain	Plane strain

Listing 6: Typesetting tables in LaTeX. Note that all columns are nicely aligned with `AlignTab` package in `Sublime Text`. There exist some softwares that can generate `Excel` tables to LaTeX or visually generate LaTeX tables online (`https://www.tablesgenerator.com`).

```latex
\begin{table}[h!]
\centering
\caption{Material parameters and characteristics for all simulations.}
\setlength\fboxsep{0pt}
\vskip-\topsep%
\smallskip%
\renewcommand\arraystretch{1.4}
\colorbox{darkgray}{%
\begin{tabularx}{0.7\textwidth}{lll}
\toprule
Parameter                & Section 5.1      & Section 5.2      \\
\midrule
Young's modulus [MPa]    & $210\times10^3$  & 145              \\
Poisson's ratio [-]      & 0.3              & 0.45             \\
Tensile strength [MPa]   & 2445             & 20               \\
    \midrule
Experimentally validated & n/a              & n/a              \\
Solver                   & multi-step AM    & single-step AM implicit-explicit \\
State                    & Plane strain     & Plane strain     \\
 \bottomrule
 \end{tabularx}%
}
\label{table:params}
\end{table}
```

Don't	Do
$\varkappa$	κ
$\hbar$	$\bar{h}$
$\underline{\underline{A}}$	$\boldsymbol{A}$ or A_{ij}

Table 5: Avoid overly complex notations.

To make this paper a complete scientific article[4], we introduce the following arbitrary equation

$$\mathscr{E}(\boldsymbol{u}, d) = \int_{\Omega_0} \left[\omega(d)\psi_0^+(\boldsymbol{\epsilon}(\boldsymbol{u})) + \psi_0^-(\boldsymbol{\epsilon}(\boldsymbol{u}))\right] \mathrm{d}V + \int_{\Omega_0} \frac{G_\mathrm{f}}{c_\alpha}\left[\frac{1}{b}\alpha(d) + b\,(\nabla d \cdot \nabla d)\right] \mathrm{d}V - \mathscr{P}(\boldsymbol{u}) \tag{4.1}$$

[4]So that beginners to LaTeX can learn how to write a complete paper.

Variable	Type	Meaning
$\mathbf{x}_p$	Vector	Particle position (time-dependent)
$\mathbf{X}_p$	Vector	Particle initial position
m_p	Scalar	Particle mass
V_p	Scalar	Particle volume
ρ_p	Scalar	Particle density
T_p	Scalar	Particle temperature
$\mathbf{P}_p$	Tensor/Matrix	Particle 1st Piola-Kirchoff stress

Table 6: Major notations can be put in a table.

For details on how to typeset formulae in LaTeX see chapter 3 of Oetiker et al. (2018).

4.5. Algorithm

The packages `algorithm and algorithmicx` can be used to typeset algorithms. See Algorithm 1 for an example. These algorithms written as pseudo codes are easier to understand than flowcharts. Yet, they are also easier to generate, directly with LaTeX. We refer to our github repository for the source.

Algorithm 1 Stress update algorithm.

Inputs: ε_p^t (equivalent plastic strain), $\boldsymbol{\sigma}_t'^{\,d}$ (un-rotated deviatoric stress), $\mathbf{d}^d$, damage D^t
Outputs: $\varepsilon_p^{t+\Delta t}$ (equivalent plastic strain), $\boldsymbol{\sigma}_{t+\Delta t}'^{\,d}$
Compute $G' = (1 - D^t)G$
$\boldsymbol{\sigma}_{\text{trial}}'^{\,d} = \boldsymbol{\sigma}_t'^{\,d} + 2G'\Delta t\mathbf{d}^d$ ▷ purely elastic stress deviator update
$\sigma_{\text{trial}}'^{\,\text{eq}} = \sqrt{\frac{3}{2}\boldsymbol{\sigma}_{\text{trial}}'^{\,d} : \boldsymbol{\sigma}_{\text{trial}}'^{\,d}}$ ▷ equivalent von Mises trial stress
$\sigma_f = \left[A + B\left(\varepsilon_p^t\right)^n\right]\left[1 + C\ln\dot{\varepsilon}_p^*\right]\left[1 - (T^*)^m\right](1 - D^t)$ ▷ JC flow stress
if $\sigma_{\text{trial}}'^{\,\text{eq}} < \sigma_f$ **then** ▷ yielding did not occur, purely elastic step
    $\boldsymbol{\sigma}_{n+1}'^{\,d} = \boldsymbol{\sigma}_{\text{trial}}'^{\,d}$ ▷ keep trial deviatoric stress
else ▷ yielding has occurred
    $\Delta\varepsilon_p = (\sigma_{\text{trial}}'^{\,\text{eq}} - \sigma_f)/(3G')$ ▷ compute the equivalent plastic strain increment
    $\varepsilon_p^{t+\Delta t} = \varepsilon_p^t + \Delta\varepsilon_p$ ▷ update the undamaged matrix plastic strain
    $\boldsymbol{\sigma}_{t+\Delta t}'^{\,d} = \frac{\sigma_f}{\sigma_{\text{trial}}'^{\,\text{eq}}}\boldsymbol{\sigma}_{\text{trial}}'^{\,d}$ ▷ scale deviatoric stress back to yield surface
end if

4.6. Source code

Source code can be included in LaTeXusing either the `Listing package or the minted package`. As the installation of the latter package is more involved (it requires an external program), we present in Listing 7 some C++ source code using the `Listing` package which does not depend on external program. We refer to the LaTeXsource of this paper for the configuration of this package to produce Listing 7.

Listing 7: Presentation of source code using the `Listing` package with the Bera Mono font (https://tug.org/FontCatalogue/beramono/).

```
#include <jive/Array.h>
#include <jem/base/System.h>
using jive::Vector;
using jive::Matrix;

Vector      a;             // double vector Array<double>
Matrix      A(10,20);      // double matrix Array<double,2>

A(slice(BEGIN,3),2) = 2.0; // third col, rows 0, 1, 2 = 2
System::out() << A << "\n";// similar to std::cout << A << "\n";
```

4.7. *Two-column format*

Preparing two-column papers is harder than one-column papers. Listing 8 presents LaTeX snippets for long figures, tables and equations that span the whole width of the paper. That's all LaTeX can do for you. For equations that are just a bit longer than one column, you have to manually modify them to make them fit.

Listing 8: Writing two-column papers using LaTeX.

```
\usepackage{mathtools, cuted} % 2 colum format, long equation

 % big figure span the whole page
\begin{figure*}[!h]
\end{figure*}

% big table span the whole page
\begin{table*}[!h]
\end{table*}

 % using strip for long equations
\begin{strip}
 % put your long equation here
\end{strip}
```

At this stage you know how to use LaTeX, to write a high quality technical paper and you have just completed one such paper. Is it ready for submission? The next section provides some steps taken to answer this question.

5. Submission

Do not submit until you are really happy with the work, as it usually does not save time to submit a piece of work which we know ourselves could be improved: the reviewers will think likewise, or be even more critical than we are ourselves of our work. If on a final read of the paper you think: "Ah. . . I could have added this study. This argument is not completely convincing to me. This graph could be better plotted. I think I forgot some relevant literature. The notations are complex, perhaps the reader will have difficulties…" Then, do not submit immediately, improve your work, and submit it when it is ready.

It is also useful to ask peers to read over your work. Before giving your first draft to your supervisor(s), have it proof-read by a peer (*i.e.*, if you are a PhD student, ask another PhD student for their opinion). This will bring the following positive points: It will value your peer as you think her/his opinion counts; It will give you insights on how understandable your paper is by someone who is connected to your field but did not do exactly the same piece of research; It will decrease the number of typos, which will enable your supervisor(s) to focus on the science as opposed to bumping over each spelling mistake, grammatical error, jargon.

It is most effective to get feedback sequentially rather than in parallel. For example, rather than asking four people to read the same version of your paper, ask one person to read the paper, then make revisions before asking the next person to read it, and so on. This prevents you from getting the same comments repeatedly.

6. Conclusions

Does this paper really need a conclusion section? Not really, but we are a bit conservative. So, here you are. Without claiming originality of ideas presented in this short paper, we have presented a collection of advices that can streamline the writing process. Following them would result in readable scientific articles which in turn save time for the authors, the editors, the reviewers and the readers.

As LaTeX and the tools we are relying on keep evolving, we will constantly update this article to reflect changes. The updated version of this paper and scripts can be obtained at the github account of the first author.

Our guidelines are rules not principles. Unlike principles, rules break all the time. Feel free to be creative as long as you write to inform not to impress. Happy writing.

Acknowledgments

The first author (V.P. Nguyen) thanks the funding support from the Australian Research Council via DECRA project DE160100577. The third author gratefully acknowledges the financial support of the Australian Research Council (ARC) Training Centre in Alloy Innovation for Mining Efficiency (IC160100036).

The guidelines presented in this paper have been formed by reading many articles and books, but mostly through interactions with our co-authors. In particular, we would like to thank L. J. Sluys and M. Stroeven (TU Delft, the Netherlands), and Chad W. Sinclair (UBC, Canada).

Appendix A Python scripts for plotting data

This appendix present two `Python` scripts to produce high quality plots which have the same font used in the paper. One can use the script shown in Listing 9 to get PDFs or the one in Listing 10 to get PGF files. PGF pic-

Table 7: A table with footnotes generated using package `threeparttable`.

Parameter	Section 5.1	Section 5.2
Young's modulus [MPa]	210×10^3	$145^{\dagger}$
Poisson's ratio [-]	0.3	0.45
Tensile strength [MPa]	2445	$20^{\ddagger}$
Experimentally validated	n/a	n/a
Solver	multi-step AM	single-step AM implicit-explicit
State	Plane strain	Plane strain

† Footnote1
‡ Taken from Mandal et al. (2019).

tures can be embedded as raw commands in LaTeX documents. The original source is https://jwalton.info/Embed-Publication-Matplotlib-Latex/.

The main idea is to get the correct figure size, and insert it in the paper without scaling. To determine the figure size, one first calculates the width of the paper, which can be done by inserting this command in the TeX file:

```
\showthe\textwidth % => width of the pdf, see in log file, is 468 pt.
```

Then you can search for 'width of the pdf' in this log file to find out the width of your paper.

Remark 3. If you are using `Matlab`, then try this script `matlab2tikz` for converting `Matlab` figures into native TikZ/Pgfplots figures. The script can be found at https://www.mathworks.com/matlabcentral/fileexchange/22022-matlab2tikz-matlab2tikz.

Appendix B Tables with footnotes

Making tables with footnotes in LaTeX is not straightforward. Table 7 is one example and we refer to the source (at our github account) to see how it was made. It is probably not an elegant solution but it is the best solution we have found.

Appendix C Reference management with BibDesk

This section presents some ideas on how we manage our library (bibliography entries plus the associated PDFs) using `BibDesk`. The original source is https://atchieu.wordpress.com (search for `BibDesk`).

We have a single `.bib` file for our library. This file is stored in a `Dropbox` folder so that it syncs to all devices. `BibDesk` is configured such that whenever a PDF is added to a bibliography entry, the PDF will be automatically renamed and stored in a specified location (in `Dropbox`) with a specified file naming convention. You do not care where they go: just go to `BibDesk` and search for the article and the PDF is there.

Listing 9: Python script for plotting data.

```python
import matplotlib as mpl
import matplotlib.pyplot as plt

def set_size(width, fraction=1, subplot=[1, 1]):
    if width == 'elsevier':
        width_pt = 468.
    elif width == 'beamer':
        width_pt = 307.28987
    else:
        width_pt = width
    fig_width_pt = width_pt * fraction       # Width of figure
    inches_per_pt = 1 / 72.27                # Convert from pt to inches
    golden_ratio = 0.75                      # (5**.5 - 1) / 2
    fig_width_in = fig_width_pt * inches_per_pt  # Figure width in inches
    fig_height_in = fig_width_in * golden_ratio * (subplot[0] / subplot[1])
    fig_dim = (fig_width_in, fig_height_in)
    return fig_dim

parser = argparse.ArgumentParser(description='Plot the evolution of variables.',
                                 add_help=False,
                                 formatter_class=argparse.RawDescriptionHelpFormatter)


parser.add_argument('--quiet', '-q', action='store_const', const=True, default=False,
                    help='do not show plot.')
args = parser.parse_args()

# Using seaborn's style
plt.style.use('seaborn-colorblind')

nice_fonts = {
        "text.usetex": True,
        "font.family": "serif",
        "axes.labelsize": 10,
        "font.size": 10,
        "legend.fontsize": 8,
        "xtick.labelsize": 8,
        "ytick.labelsize": 8,
}

mpl.rcParams.update(nice_fonts)

# Dimer 5: Read Data
fnames = ['log.mpm', 'dam-break-Sun.csv', 'water-break-experiment.csv']
dat    = [pylab.genfromtxt(f,skip_header=1) for f in fnames]

fig, ax = plt.subplots(1, 1, figsize=set_size('elsevier', fraction=0.6))
# fig, (ax1,ax2) = plt.subplots(1, 2, figsize=set_size(width, subplot=[1, 2]))

for i, f in enumerate(fnames):
    x = dat[i][:,3]
    y = dat[i][:,4]
    ax.plot(x, y, color='black', label='MPM',linewidth=1.5, linestyle='dotted')

    xmax = max(xmax, np.max(x))
    ymax = max(ymax, np.max(y))

ax.set_xlabel(r'$T$')
ax.set_ylabel(r'$L(T)$')
ax.set_ylim(1.0, 2.8)
ax.set_xlim(0, 1.6)
ax.legend(loc=0)
plt.tight_layout()
plt.savefig('./water-break-plot.pdf', format='pdf', bbox_inches='tight')

if args.quiet == False:
    plt.show()
```

References

M. Ashby. How to write a paper. *Engineering Department, University of Cambridge, Version*, 5:38, 2000. [Cited on pages 1, 1, and 10]

Listing 10: Python script for plotting data.

```
nice_fonts = {
        "text.usetex": True,
        "font.family": "serif",
        "pgf.texsystem": "pdflatex",
        "font.family": "serif",
        "font.serif": [],
        "font.sans-serif": [],
        "font.monospace": [],
        "pgf.preamble": [
        # put LaTeX preamble declarations here
        r"\usepackage[utf8x]{inputenc}",
        r"\usepackage[T1]{fontenc}",
        # macros defined here will be available in plots, e.g.:
        r"\newcommand{\vect}[1]{#1}",
        # You can use dummy implementations, since you LaTeX document
        # will render these properly, anyway.
        ],
        "axes.labelsize": 10,
        "font.size": 10,
        "legend.fontsize": 8,
        "xtick.labelsize": 8,
        "ytick.labelsize": 8,
}
plt.savefig('./cold-spray-plots.pgf', bbox_inches='tight')
if args.quiet == False:
    plt.show()
```


R. A. Day. How to write and publish scientific papers, 1998. [Cited on pages 1, 1, and 10]

A. de Vaucorbeil, V. P. Nguyen, S. Sinaie, and J. Y. Wu. Material point method after 25 years: theory, implementation and applications. Submitted to Advances in Applied Mechanics, 2019. [Cited on page 9]

J. B. C. Engelen. How to include an SVG image in LaTeX, 2010. URL http://mirrors.ibiblio.org/CTAN/info/svg-inkscape/InkscapePDFLaTeX.pdf. [Cited on page 15]

T. W. Fawcett and A. D. Higginson. Heavy use of equations impedes communication among biologists. *Proceedings of the National Academy of Sciences*, 109(29):11735–11739, 2012. ISSN 0027-8424. doi: 10.1073/pnas.1205259109. URL https://www.pnas.org/content/109/29/11735. [Cited on pages 17 and 18]

G. D. Gopen and J. A. Swan. The science of scientific writing. *American Scientist*, 78(6):550–558, 1990. [Cited on pages 5 and 5]

T. Gray. *Publish and flourish: become a prolific scholar.* Teaching Academy, 2005. [Cited on page 11]

A. D. Higginson and T. W. Fawcett. Comment on 'are physicists afraid of mathematics?'. *New Journal of Physics*, 18(11):118003, nov 2016. [Cited on pages 17 and 18]

R. L. Iles. *Guidebook to better medical writing*. Iles publications, 1997. [Cited on page 1]

S. P. Jones. How to write a great research paper, 2016. URL https://www.microsoft.com/en-us/research/academic-program/write-great-research-paper/. [Cited on page 4]

T. K. Mandal, V. P. Nguyen, and J.-Y. Wu. Length scale and mesh bias sensitivity of phase-field models for brittle and cohesive fracture. *Engineering Fracture Mechanics*, 217(106532), 2019. [Cited on pages 7 and 23]

B. Oakley, T. Sejnowski, and A. McConville. *Learning How to Learn: How to Succeed in School Without Spending All Your Time Studying; A Guide for Kids and Teens*. TarcherPerigee, 2018. [Cited on page 11]

T. Oetiker, H. Partl, I. Hyna, and E. Schlegl. The not so short introduction to LaTeX, 2018. URL http://tug.ctan.org/info/lshort/english/lshort.pdf. [Cited on page 20]

C. E. Paiva, J. P. d. S. N. Lima, and B. S. R. Paiva. Articles with short titles describing the results are cited more often. *Clinics*, 67(5):509–513, 2012. [Cited on page 10]

K. W. Plaxco. The art of writing science. *Protein science: a publication of the Protein Society*, 19(12):2261, 2010. [Cited on pages 1 and 10]

A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle. A material point method for snow simulation. *ACM Transactions on Graphics*, 32(4):1,

2013. [Cited on page 8]