

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

Vinh Phu Nguyen<sup>a,\*</sup>, Stephane Bordas<sup>b</sup>, Alban de Vaucorbeil<sup>c</sup>

<sup>a</sup>Department of Civil Engineering, Monash University, Clayton 3800, VIC, Australia

<sup>b</sup>Institute of Computational Engineering, University of Luxembourg, Faculty of Sciences Communication and Technology, Luxembourg

<sup>c</sup>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 has been 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 pain. We hope it works for you too. Due to our personal preferences, the discussion is confined to L<sup>A</sup>T<sub>E</sub>X based typesetting where we present guidelines to prepare high-quality images, tables, algorithms and source codes using L<sup>A</sup>T<sub>E</sub>X. We also discuss writing guidelines such as paper structure, major components of a paper from title, abstract, introduction, results to references, and some common mistakes.

*Keywords:* scientific writing; L<sup>A</sup>T<sub>E</sub>X; scientific publication; high quality writing; reproducibility.

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Tools</b>	<b>4</b>
<b>3</b>	<b>Writing tips</b>	<b>6</b>
3.1	General guidelines . . . . .	6
3.2	How to structure your paper . . . . .	8
3.2.1	Title . . . . .	9
3.2.2	Abstract . . . . .	9
3.2.3	Global paragraph for long sections . . . . .	10
3.2.4	Introduction section . . . . .	10
3.2.5	Results section . . . . .	11

---

\*Corresponding Author

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

3.2.6	Conclusion section	13
3.2.7	Appendix and footnotes	14
3.2.8	References	14
3.3	Notations and equations	15
3.4	Figures	16
3.5	Some common mistakes	17
3.6	Writing process: an iterative process	18
<b>4</b>	<b>L<sup>A</sup>T<sub>E</sub>X tips</b>	<b>19</b>
4.1	Packages	19
4.2	Figures	20
4.2.1	Graphs	21
4.2.2	Sketches	21
4.2.3	Contour plots	22
4.3	Tables	24
4.4	Equations	24
4.5	Algorithm	25
4.6	Source code	27
4.7	Two-column format	27
<b>5</b>	<b>Submission</b>	<b>27</b>
<b>6</b>	<b>Conclusions</b>	<b>28</b>
	<b>Appendices</b>	<b>29</b>
	<b>Appendix A Python scripts for plotting data</b>	<b>29</b>
	<b>Appendix B Tables with footnotes</b>	<b>29</b>
	<b>Appendix C Reference management with BibDesk</b>	<b>31</b>

## 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 advice 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 software 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 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 writing tools are presented. These tools are mostly open source and cross platform. Then, Section 3 provides actionable suggestions on how to write a good paper. Particularly, we present an iterative writing process that is intertwined with your research. The most important recommendation is to write early in the research process instead of waiting until all results are ready. 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 papers 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. We were among them. This is not surprising since we, as engineers, rarely change something that is working. 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 University – saw it: ‘scientific writing

is bad writing'. Or as a paper published in Nature Index concluded 'Science is getting harder to read' <sup>1</sup>.

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 has been 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 writing 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.

---

<sup>1</sup><https://www.natureindex.com/news-blog/science-research-papers-getting-harder-to-read-acronyms-jargon>.

- `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 that is 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 the 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. The co-authors can still use their own favorite `LATEX` editor<sup>2</sup> (e.g. the first author used `Sublime Text` while the last author used `Emacs`). We find this organization to be better than `Overleaf` at <https://www.overleaf.com>, which forces all authors to use a web-based `LATEX` editor.

*Remark 1.* Without going into 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.

```

momentum in time
is equal to the sum of all external forces (volume and surface
forces) acting on the body. It is described by the so-called \emph{the momentum equation}:

\begin{equation}\Label{chap2-linear-momentum-eq}
\rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \mathbf{\sigma} + \rho \mathbf{b}
\quad \text{or} \quad \rho \dot{v}_i = \sigma_{ji,j} + \rho b_i
\end{equation}

\frac{Dv}{Dt} = \nabla \cdot \sigma + \rho b \quad \text{or} \quad \rho \dot{v}_i = \sigma_{ji,j} + \rho b_i

```

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.

<sup>2</sup>which is most often also their own coding editor.

### 3. Writing tips

This section presents writing tips. We start with a presentation of general guidelines in Section 3.1. Then we discuss the paper structure in Section 3.2: nearly all major components of a paper are treated: title, abstract, introduction, results, conclusions, appendices, and references. Next, we discuss equations in Section 3.3 and figures in Section 3.4. Some common mistakes and a few tips to make your paper less verbose or wordy are given next in Section 3.5. Finally, Section 3.6 – the most significant contribution of the paper – outlines an iterative writing process in which writing is intertwined with other research activities. Recently, we realized that this technique has been, not surprisingly, adopted by other researchers, *e.g.* 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)<sup>3</sup>;
5. Avoid jargon and abbreviations;
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. Another example is, from S.P. Jones (yes, the same Jones who has written the interesting paper on how to write great papers (Jones, 2016))

---

<sup>3</sup>Watch also the interesting talk of Judy Swan at <https://www.youtube.com/watch?v=1pzjxYCwb08>.

*Security proofs* of cryptographic protocols are crucial for the security of everyday electronic communication. However, *these proofs* tend to be complex and difficult to get right. To make it easier to manage *such proofs*, Jones et al. have proposed a new design principle, called the *game-playing technique*. *This technique* follows a code-based approach where the security properties are formulated in terms of probabilistic programs, called games.

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 5 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 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.

Consider the following sentence appearing in an abstract:

*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. Reproducibility requires that findings can be verified independently. 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. The bottom line is: *Explain everything*; if something is obvious to you does not mean that it is obvious to your reader. 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*

Typically a paper in the field of computational of computational engineering and sciences consists of the following parts:

- Title
- Abstract
- Introduction
- Methods
- Results
- Conclusions
- Acknowledgments
- Appendices
- References

In this section we discuss some suggestions on how to write some of these parts. We start with titles in Section 3.2.1, which is followed by Section 3.2.2 on the abstract part. Then we present a way to organize long sections by having a global paragraph before going into its subsection (Section 3.2.3). Next, we discuss how to write a compelling introduction section (Section 3.2.4), and what the conclusion part should include (Section 3.2.6). Appendices and footnotes are discussed in Section 3.2.7 and references in Section 3.2.8.

An excellent article on how to structure a scientific paper can be found at <https://www.nature.com/scitable/topicpage/scientific-papers-13815490/>. There are certainly overlaps between it and our discussion.

### 3.2.1. Title

Why should you pay more attention to the title of your paper? This is because

- the title is the part of a paper that is read the most;
- it is usually read first; and thus a good title gives a good first impression;
- papers with short titles got more citations (Paiva et al., 2012);
- using a question mark in a paper's title reduces the citations;
- using a colon tended to improve the citations;

What a good title look like then? It should

- Indicate accurately the subject/scope of the study;
- not contain abbreviations;
- not include phrases such as "study of," "analysis of" or similar constructions.

The following title

*An efficient and robust staggered algorithm applied to the quasi-static description of brittle fracture by a phase-field approach*

can be modified to make it better

*An efficient and robust staggered solver for a phase-field model of quasi-static brittle fracture*

### 3.2.2. Abstract

The abstract of a paper is the most important section of your paper. This is so because it is the second section that is read by journal editors (they read your cover letter first). Furthermore, once your paper has been published, its abstract is the first section that is clearly examined by readers. And in many cases, it is the only section of the manuscript that will be ever read. Sadly, many authors write the abstract in a great rush, almost as an afterthought (probably too eager to let the paper go?).

The abstract should be a concise standalone summary of your paper. It must include the background, gaps, methodology and results. In other words, it answers these questions: Why did you do the study? What did you do? What did you find? What did you conclude? Fig. 2 presents an example of a good abstract. We label it a good abstract because it answers the above questions and it does not contain abbreviations.

---

## Abstract

In this paper, we investigate fracture in shells with a phase-field modeling approach. The shell model is based on solid-shell kinematics with small rotations and displacements and is discretized using quadratic Non-Uniform Rational B-Spline basis functions. Membrane and shear locking is alleviated through the Assumed Natural Strain approach. The solid-shell formulation is combined with a brittle phase-field model for elastic materials, as well as with a ductile fracture model for elasto-plastic materials exhibiting  $J_2$  plasticity with isotropic hardening. Several examples demonstrate the ability of the proposed framework to capture crack initiation, propagation, merging and branching phenomena as well as crack bulging effects in shells under different states of loading.

© 2016 Elsevier B.V. All rights reserved.

*Keywords:* Brittle fracture; Ductile fracture; Phase-field modeling; Solid shells; Isogeometric analysis; Assumed natural strains

---

Figure 2: Example of a good abstract, taken from [Ambati and Lorenzis \(2016\)](#).

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

### 3.2.3. 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. The idea is to prepare your readers for the structure ahead at all levels; not doing so might make your reader lost. See Fig. 3 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 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. We first saw this technique probably in [Huang et al. \(2003\)](#). Furthermore, a table with all parameters used for all simulations would be helpful, see *e.g.* Table 6, as it is harder to get them if they are embedded in the text.

### 3.2.4. 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.

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

What we commonly see in the literature 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  $\mathbf{u}$  and damage field  $d$  are minimizers of the following total energy functional of the solid

$$\mathcal{E}(\mathbf{u}, d) = \int_{\Omega_0} [\omega(d)\psi_0^+(\boldsymbol{\epsilon}(\mathbf{u})) + \psi_0^-(\boldsymbol{\epsilon}(\mathbf{u}))] dV + \int_{\Omega_0} \frac{G_f}{c_\alpha} \left[ \frac{1}{b}\alpha(d) + b(\nabla d \cdot \nabla d) \right] dV - \mathcal{P}(\mathbf{u})$$

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  $\psi_0^+(\boldsymbol{\epsilon}(\mathbf{u}))$  and  $\psi_0^-(\boldsymbol{\epsilon}(\mathbf{u}))$ , respectively.

model	$\alpha(d)$	$\omega(d)$	fracture type	length-scale	sup.	Parameters
AT2	$d^2$	$(1-d)^2$	brittle	$b = (27/256)l_{\text{ch}}$	$\infty$	$E_0, \nu_0, G_f, b$
AT1	$d$	$(1-d)^2$	brittle	$b = (3/8)l_{\text{ch}}$	$4b$	$E_0, \nu_0, G_f, b$
PF-CZM	$2d - d^2$	$\frac{(1-d)^p}{(1-d)^p + Q(d)}$	brittle/cohesive	numerical param.	$\pi b$	$E_0, \nu_0, G_f, f_t$

One way to visually demonstrate the contributions of your paper is to use a table in which a comparison with previous models is given. We borrowed this idea from the computer graphics community, see e.g. [Stomakhin et al. \(2013\)](#). We have adopted this idea in [de Vaucorbeil et al. \(2019\)](#) where we have provided a table (Table 2) where we compared different variants of the material point method (MPM).

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. 4). In this way, you increase the chance that your paper gets read to the end.

### 3.2.5. Results section

In many fields a paper contains one section discussing the results and one section “Discussions” to elaborate on the results. However, in the field of computational mechanics, most often these two are combined into one section named

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.

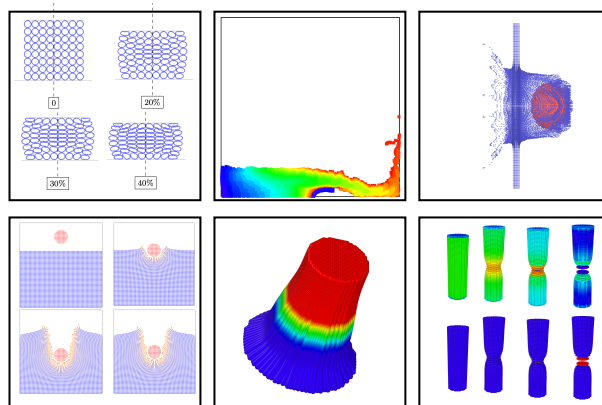


Figure 4: A figure with all good results that the model in your paper can deliver can help to intrigue readers so that they will read your paper to the end. Taken from [de Vaucorbeil et al. \(2019\)](#).

“Numerical examples” or the likes.

As we have discussed in Section 3.2.3 for long and complex section we need to have a global paragraph planning the reader. For the Results section, this global paragraph should explain: (1) what examples are considered and for what purposes, (2) which codes are used and (3) what not considered and reason of doing so. If all these details are given it is easier for the reviewers and the researchers transitioning to a new field to know what is going on. Fig. 5 presents such a global paragraph.

For each example, before presenting the results, we should also help the reader by stating what is the question that we want to find answer with it, and what qualitative and quantitative measures are used for that purpose. For example, for example 1, we can write:

*The aim of this example is to verify if the proposed method provides mesh convergent results. To this end, we consider four finite element meshes of increasing number of elements, and for each mesh we monitor the load-displacement measured at point P (see Fig.1). The method is said to be mesh convergence if the load-displacement plots converge for the used meshes.*

### 3. Examples

To demonstrate the performance of the generalized particle in cell method, we present the following tests

- Test 1: rubber block in a gravity field (Section 3.1)
- Test 2: rubber block hitting a rigid wall (Section 3.2)
- Test 3: collision of two compressible Neo-Hookean rings (Section 3.3)
- Test 4: Taylor anvil test (Section 3.4)
- Test 5: a sphere penetrates a target (Section 3.5)
- Test 6: bi-material cylinder under pressure (Section 3.6)

All these tests cover a wide range of deformation modes. Test 1 is for testing GPIC for problems exhibiting massive tensile deformation. Test 2 is for compressive deformation, Test 3 is for collision that involves both tensile and compressive deformation. Test 4 is the well known Taylor anvil test with large plastic deformation. Test 5 is a 3D penetration problem. Test 6 is an elasticity problem involving material interfaces and Neumann boundary condition, both of which are difficult to treat in a particle method.

Figure 5: A global paragraph planning the reader for the Results section.

#### 3.2.6. 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 written conclusion section should include the following items:

- One or two sentences summarizing what the paper has been about;
- A summary of the key findings of the paper;
- A list of potential improvements.

The key findings can be presented using bullet points. And as far as bullet points are concerned, we have to mention parallel structure. Parallel structure means that coordinate parts of a sentence, such as items in a series or list, have the same grammatical form.

For example, in a draft version, one student of ours wrote (in the introduction section): We present the following concrete contributions:

- A unified fourth order phase field fracture framework for brittle and quasi-brittle solids which can be considered as an extension of the unified PFM of Wu (2017);
- A semi-analytical (analytical-numerical) approach for PF-CZM;
- A almost length scale insensitive fourth order PFM;

- The first fourth order PFM for cohesive fracture;
- The PF-CZM is applied to study the phenomena of crack kinking in anisotropic brittle fracture

We do not need to be a native English to recognize that the above list is wrong. Just by reading it out loud we can see that the first three bullet points are ok, but something wrong with the fourth and fifth bullets.

Using parallel structure, the above list must be written as follows: We present the following concrete contributions:

- A unified fourth order phase field fracture framework for brittle and quasi-brittle fracture;
- A semi-analytical approach for the fourth order PF-CZM;
- A comparison of the fourth order PF-CZM against the second order PF-CZM;
- A study of the phenomena of crack kinking and sawtooth cracking in strongly anisotropic brittle fracture.

### 3.2.7. 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 3.3, 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.

### 3.2.8. 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 too 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. Notations and equations

Avoid overly complex notations, see Table 3 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 4 does a good job to introduce main notations.

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

Table 3: Avoid overly complex notations.

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
$\rho_p$	Scalar	Particle density
$T_p$	Scalar	Particle temperature
$\mathbf{P}_p$	Tensor/Matrix	Particle 1 <sup>st</sup> Piola-Kirchoff stress

Table 4: Major notations can be put in a table.

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 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.

A paper in the field of computational sciences certainly contains some mathematical equations and/or expressions. To demonstrate how equations are written, we use the following one (taken from one of our papers)

$$\mathcal{E}(\mathbf{u}, d) = \int_{\Omega_0} [\omega(d)\psi_0^+(\boldsymbol{\epsilon}(\mathbf{u})) + \psi_0^-(\boldsymbol{\epsilon}(\mathbf{u}))] dV + \int_{\Omega_0} \frac{G_f}{c_\alpha} \left[ \frac{1}{b}\alpha(d) + b(\nabla d \cdot \nabla d) \right] dV - \mathcal{P}(\mathbf{u}) \quad (3.1)$$

where the first term is the strain energy, the second term is the surface energy and the third term the external energy.

Collectively, they define the energy of the mechanical system  $\mathcal{E}(\mathbf{u}, d)$ <sup>4</sup>.

Note that we have "voiced out" the above equation in "words" so that the reader will have one more way of understanding, and the equations will not look as blunt and intimidating, in particular when you have a lot of notation which does not appear in literature. Later on we refer to this equation using its number as: *The energy functional in Equation (3.1) is minimized to obtain ...*

We do not want to discuss about punctuations in equations as this problem is still a debate. Do whatever you feel comfortable with. But the editor can ask you to add punctuations in your equations!

### 3.4. Figures

Figures are an important element of reporting the findings of your research. The reader usually starts by reading the abstract, conclusion and figures. If the figures are self-contained and convincing, it is likely that the reviewers will accept the paper without asking a large number of questions. If they have to read the whole text to understand the worth of each figure, this will slow their progress and could make them impatient, hence decrease the chance of giving recommendation for your paper.

A high quality figure is worth a thousand words. By a high quality figure we meant the one that has a legible font size, a high resolution, is color-blindness aware, of which all the axes are clearly defined, those sorts of things. Later on we will discuss how to prepare such figures (Section 4.2). We emphasize the importance of high quality figures because we have seen so many figures which are, without better words, of very low quality.

Tables and figures, although important components of any research papers, are just that—components; you can publish a paper without them but you cannot publish tables or figures without a paper. Thus a figure (or table) exists to support the text. So the first rule is: make sure that every table and figure is mentioned in the text. Even though there are some  $\LaTeX$  packages (*e.g.* Refcheck) that help to find out unused figures and tables, our practice is when we insert a figure in our document, we cite it immediately. And the second rule is: remove any figure if it does not support your discussion. The third rule is good figure captions. 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. And the final is how figures/tables are referred to in the text.

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 (including our papers). A better version of the above is:

---

<sup>4</sup>Any mathematical symbols appearing in the text must be typeset using a math font. In  $\LaTeX$  this is achieved by putting the symbols in two dollar signs.

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

We say that this is a better version because it did not use the phrase ‘As can be seen’. It is clear that the use of too many such phrases should (and can be) be avoided. Actually we already use this style when we cite published articles and books to support an argument. We can do the same thing with figures and tables.

### 3.5. Some common mistakes

Some common mistakes are given in Table 5. 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
<b>The</b> Table/Figure 2	Table/Figure 2
<b>The</b> Equation (2.2)	Equation (2.2)
<b>The</b> 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 <b>long</b> time	... for more than 20 years
A <b>bad</b> result	A poor/negative result
This section <b>serves</b> to explain	This section explains
It is <b>obvious/clear</b> ...	
<b>Due to the fact that</b> ...	Because ...
<b>It should be noted that</b> there are 5 samples in this study	This study consisted of 5 ...
<b>In order to</b> include ...	To include ...
The difference was <b>found to be</b> significant	The difference was significant
We plotted the data <b>by</b> using ...	We plotted the data using ...
Utilize or usage	Use
We <b>think/believe/feel</b> that the results are good	The results are good
<b>Existing</b> works ...	Previous work
Using adjectives such as ‘very’, ‘always’, ‘never’	
Using words like ‘ground-breaking’, ‘paradigm shift’	
Using ‘Above-mentioned’ or ‘aforementioned’	
Use long titles	Always spell out an acronym the first time it is used Use short titles (Paiva et al., 2012) Use a spell checker to get rid of all spell errors

Table 5: Some common mistakes. A good way to improve your writing is by reading your writing, including all of the writing out loud. Your ears can often pick out sentence fragments and grammatical errors better than your eyes. If you find yourself saying a series of fragmented sentences or feeling something unnatural, you should do some rewriting.

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.

### 3.6. Writing process: an iterative process

The first idea of the presented writing process 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. 6.

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 T<sub>E</sub>X 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 become more 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 T<sub>E</sub>X 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. T<sub>E</sub>X keywords are highlighted in **bold blue font**.

```
1 \documentclass[authoryear,3p,times,preprint,review,fleqn]{elsarticle}
2 \title{\textbf{}}
3 \begin{abstract}
4 \end{abstract}
5 \tableofcontents % TOC
6 \section{Introduction}
7 \section{Methodology}
8 \section{Examples}
9 \section{Conclusions}
10 \section*{Acknowledgments}
11 \bibliographystyle{abbrvnat} % bib style
12 \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 comes. Jotting down the idea on a phone and you're done with this part of the paper.

While working on a paper, you read the literature (we always read it anyway). If you find a good paper relevant to

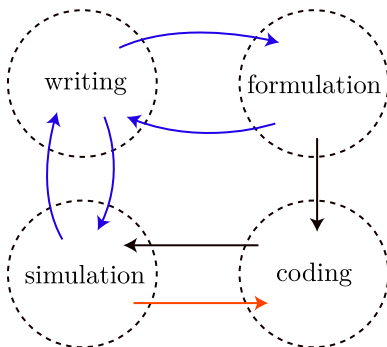


Figure 6: Writing is intertwined with other activities of the project.

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 (or JabRef) 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. That is why we call this writing process an effortless experience. 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  $\text{\LaTeX}$  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  $\text{\LaTeX}$ .

#### 4. $\text{\LaTeX}$ tips

We present in this section some  $\text{\LaTeX}$  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, 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  $\text{\LaTeX}$  skills. Listing 2 provides an updated list of  $\text{\LaTeX}$  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.

```

1 \usepackage{amsmath,amssymb, mathtools, mathrsfs, stmaryrd, titletoc}
2 \usepackage{natbib}
3 \usepackage[scaled=0.92]{helvet} % set Helvetica as the sans-serif font
4 \renewcommand{\rmdefault}{ptm} % set Times as the default text font
5 \usepackage[retainorgcmds]{IEEEtrantools}
6 \usepackage[usenames]{color}
7 \usepackage{tabularx} % tables
8 \usepackage{booktabs} % better tables
9 \usepackage{multirow} % multi-row tables
10 \usepackage[font=small, labelfont=md]{caption, subfig} % sub-figures (see Fig. 6)
11 \usepackage[T1]{fontenc} % typing french
12 \usepackage[bookmarks=true, colorlinks=true, linkcolor=blue, backref=page]{hyperref}
13 \usepackage{float} % make new float environment such as boxes (captioned)
14 \usepackage{listings} % insert source code, used herein to insert LaTeX and Python codes
15 \usepackage{algorithm} % flow wrapper for algorithm, similar to block commands table/figure
16 \usepackage{algpseudocode} % second algorithm typesetting environment (actual pseudocode)
17 %microtype: for Micro-Typographic Improvements
18 \usepackage[activate={true, nocompatibility}, final, tracking=true, kerning=true,
19 spacing=true, factor=1100, stretch=10, shrink=10]{microtype}
20 \usepackage{nicefrac} % type inline fractions: \nicefrac{1}{2}
21 \usepackage{numprint} % \numprint{10000} => 10 000 not 10000
22 \usepackage[title, titletoc, toc]{appendix}
23 \usepackage[capitalise]{cleveref} %Basically, cleveref must be loaded last.
24 \definecolor{darkgray}{rgb}{0.95,0.95,0.95} % color used in tables
25 % cleverref package: just do \cref{label} for figures, tables, equations anything
26 % the package will determine the correct prefix be it Fig. or Equation or Listing.
27 \crefname{figure}{Fig.}{Figs.}
28 \crefname{equation}{Equation}{Equations}
29
30 \renewcommand*{\backref}[1]{}
31 \renewcommand*{\backrefalt}[4][[%
32 \ifcase #1 %
33 \or Cited on page~#2%
34 \else Cited on pages #2%
35 \fi%
36 ]}]

```

## 4.2. Figures

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). For more detail on preparing figures, we refer to Rougier et al. (2014).

### 4.2.1. Graphs

It is not a requirement that the font used in graphs (e.g. bar charts, error charts,  $x$ - $y$  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. 7) or PGFs (Portal Graphic Format) with matching font (see Fig. 8). The  $\LaTeX$  code used to include this type of plot is shown in Listing 3. We refer to Listings 10 and 11 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. Each figure should have a unique label (here is 'fig:cold'), which is used to refer to it in text.

```

1 \begin{figure}[!ht]
2   \centering
3   % only one of the following three
4   \includegraphics{cold-spray-plots.pdf} % insert a PDF
5   \input{cold-spray-plots.pgf}          % insert a PGF
6   \input{output.pdf_tex}                % insert a pdf_tex
7   \caption{Cold spraying with a single impact: evolution of plastic strain and temperature.}
8   \label{fig:cold}                       % this label is used in \cref{fig:cold}
9 \end{figure}

```

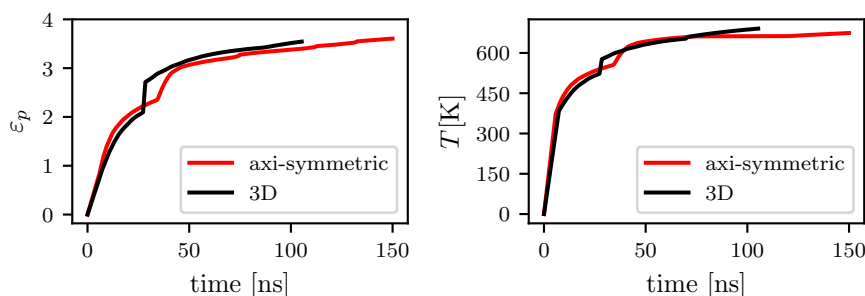


Figure 7: A PDF figure of which the font nearly matches the text font: evolution of plastic strain  $\varepsilon_p$  and temperature  $T$  in time. Symbols, if any, in figures should be typeset with  $\LaTeX$ . Be thoughtful about color blindness that affects around 8% of men, particularly an inability to distinguish red and green. `matplotlib` can be color-blind appropriate, see line 29 of Listing 10. 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. 9 is a collection of 4 figures, with caption for each one of them. The corresponding  $\LaTeX$  code is given in Listing 4.

### 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`<sup>5</sup> to typeset whatever symbols and drag and drop them to `Illustrator` as embedded PDFs<sup>6</sup>. Fig. 10a presents an example. The same thing can be done using `Inkscape` with some  $\LaTeX$  extensions. This

<sup>5</sup>This software can be found at <https://www.chachatelier.fr/latexit/>. Note that, it works only for Mac OS.

<sup>6</sup>Presentations can be created using `Keynotes` with equations created using `LaTeXiT` in exactly the same way.

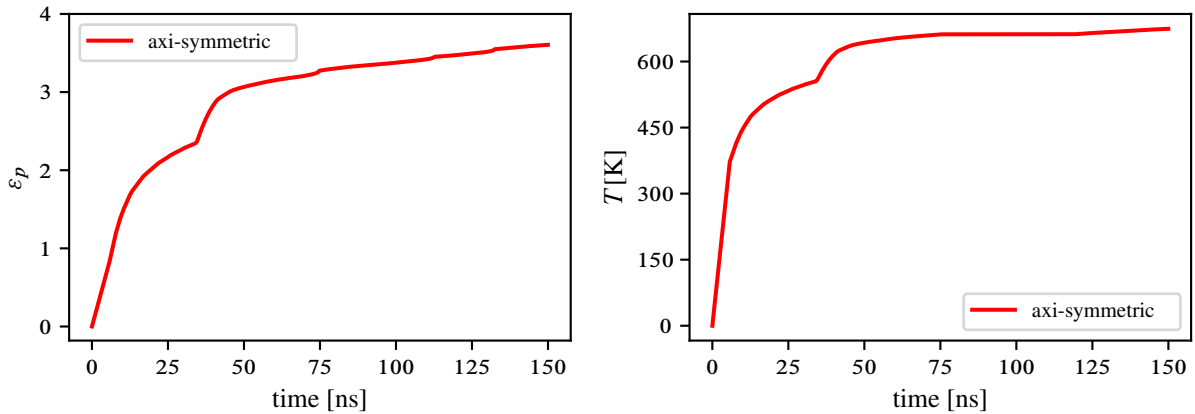


Figure 8: A PGF figure of which the font matches the text font: evolution of plastic strain  $\varepsilon_p$  and temperature  $T$  in time. PGF pictures are embedded as raw commands in  $\LaTeX$  documents and thus can slow down the compilation process (*i.e.*, the process from a  $\TeX$  file to the final PDF).

is achieved by writing all text or formula using  $\LaTeX$  syntax in Inkscape. For example, if we want to get  $\nabla\sigma$ , we write the following text in Inkscape:

```
\nabla \sigma
```

Saving the drawing into the SVG format (SVG is short for scalable vector graphic file), and then exporting it as a PDF\_TEX file using the following command (Engelen (2010))<sup>7</sup>:

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

This command generates two files: one is *output.pdf* and the other is *output.pdf\_tex*. And then we insert the figure using the file *output.pdf\_tex*, see Listing 3 (use line 5). Fig. 10b 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. 11. 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. 9), which are the outcomes of some visualization applications such as ParaView<sup>8</sup>. 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

<sup>7</sup>This can also be done using the GUI: File/Save as/ and enable the option 'Omit text in PDF and create LaTeX file'.

<sup>8</sup>As we embrace as much as possible open source software, we did not mention tecplot, a good data visualization package developed explicitly for engineers and scientist.

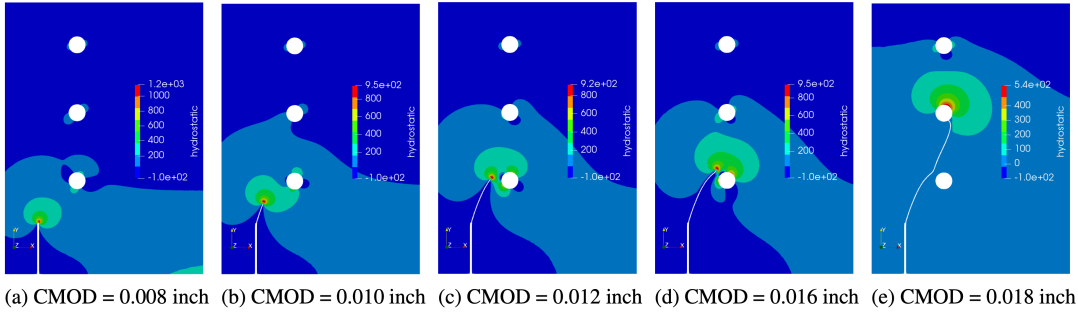


Figure 9: Stacking multiple pictures together with sub-captions using the package subfig.

Listing 4: Stacking multiple images using  $\text{\LaTeX}$  package subfig.

```

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

```

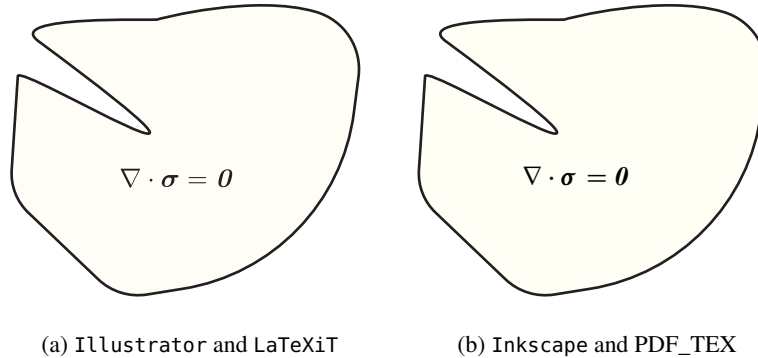


Figure 10: Using Illustrator and Inkscape to produce vector images with  $\text{\LaTeX}$  symbols. The font in figure (a) is slightly different from the one in the text ( $\nabla \cdot \sigma = 0$ ), while it matches perfectly in figure (b).

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.

For complex drawings we have to combine the different tools together. Fig. 12 shows one example in which we combined ParaView, matplotlib and Inkscape.

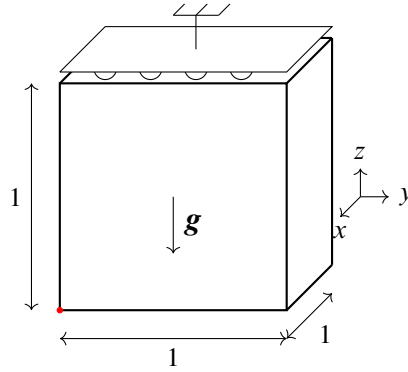


Figure 11: Example of schematic made using TikZ.

### 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 6 satisfies all the criteria. The  $\LaTeX$  code is shown in Listing 6. Table 7 in Appendix B presents a table with footnotes.

Table 6: Material parameters and characteristics for all simulations.

Parameter	Section 5.1	Section 5.2
Young's modulus [MPa]	$210 \times 10^3$	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

### 4.4. Equations

With  $\LaTeX$  it is possible to get nice mathematical symbols and equations, for example

$$\mathcal{E}(\mathbf{u}, d) = \int_{\Omega_0} [\omega(d)\psi_0^+(\boldsymbol{\epsilon}(\mathbf{u})) + \psi_0^-(\boldsymbol{\epsilon}(\mathbf{u}))] dV + \int_{\Omega_0} \frac{G_f}{c_\alpha} \left[ \frac{1}{b}\alpha(d) + b(\nabla d \cdot \nabla d) \right] dV - \mathcal{P}(\mathbf{u}) \quad (4.1)$$

The TeX code used to generate this equation is given in Listing 7. For details on how to typeset formulas in  $\LaTeX$  see chapter 3 of Oetiker et al. (2018).

Listing 5: TikZ code used to obtain Fig. 11.

```

1 \begin{tikzpicture}[darkstyle/.style={circle,draw,fill=gray!40,minimum size=20}]
2   \tikzmath{
3     \L = 3;
4     \h = \L / 30;
5   }
6
7   \draw [thick] (-\L/2,-\L/2) rectangle (\L/2,\L/2);
8   \draw [thick] (\L/2,-\L/2) -- (\L/2+\L/5,-\L/2+\L/5);
9   \draw [thick] (\L/2+\L/5,-\L/2+\L/5) -- (\L/2+\L/5,\L/2+\L/5);
10  \draw [thick] (\L/2,\L/2) -- (\L/2+\L/5,\L/2+\L/5);
11  \draw [thick] (-\L/2+\L/5,\L/2+\L/5) -- (\L/2+\L/5,\L/2+\L/5);
12  \draw [thick] (-\L/2,\L/2) -- (-\L/2+\L/5,\L/2+\L/5);
13
14  \draw [->] (0,0) -- (0,-\L/4);
15  \node [right,thick] at (0,-\L/8) {\large $\boldsymbol{g}$};
16
17  \draw [<->] (-\L/2,-\L/2 - \L/8) -- (\L/2,-\L/2 - \L/8);
18  \draw [<->] (\L/2,-\L/2 - \L/8) -- (\L/2+\L/5,-\L/2 - \L/8+\L/5);
19  \draw [<->] (-\L/2 - \L/8, - \L/2) -- (-\L/2 - \L/8, \L/2);
20  \node [anchor=east] at (-\L/2 - \L/8,0) {1};
21  \node [anchor=north] at (0,-\L/2 - \L/8) {1};
22  \node [anchor=north west] at (\L/2+\L/10,-\L/2 - \L/8+\L/10) {1};
23  \draw [] (-\L/10,\L/2 + \L/15) circle (\L/20);
24  \draw [] (-\L/5-\L/10,\L/2 + \L/15) circle (\L/20);
25  \draw [] (\L/10,\L/2 + \L/15) circle (\L/20);
26  \draw [] (\L/5+\L/10,\L/2 + \L/15) circle (\L/20);
27  \draw [-,fill=white] (-\L/2,\L/2 + \L/20) -- (\L/2,\L/2 + \L/20) --%
28  (\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);
29  \draw [-] (\L/10,\L/2 + \L/20+\L/10) -- (\L/10,\L/2 + 3*\L/10);
30  \draw [-] (\L/10-\L/10,\L/2 + 3*\L/10) -- (\L/10+\L/10,\L/2 + 3*\L/10);
31  \draw [-] (\L/10-\L/10,\L/2 + 3*\L/10) -- (\L/10-\L/10 + \L/20,\L/2 + 3*\L/10 + \L/20);
32  \draw [-] (\L/10,\L/2 + 3*\L/10) -- (\L/10 + \L/20,\L/2 + 3*\L/10 + \L/20);
33  \draw [-] (\L/10+\L/10,\L/2 + 3*\L/10) -- (\L/10+\L/10 + \L/20,\L/2 + 3*\L/10 + \L/20);
34
35  % Triad:
36  \draw [->] (\L/2+\L/5+\L/8,0) -- (\L/2+\L/5+2*\L/8,0);
37  \draw [->] (\L/2+\L/5+\L/8,0) -- (\L/2+\L/5+\L/8*0.29,0-0.71*\L/8);
38  \draw [->] (\L/2+\L/5+\L/8,0) -- (\L/2+\L/5+\L/8,0+\L/8);
39  \node [anchor=north] at (\L/2+\L/5+\L/8*0.29,0-0.71*\L/8) {$x$};
40  \node [anchor=south] at (\L/2+\L/5+\L/8,0+\L/8) {$z$};
41  \node [anchor=west] at (\L/2+\L/5+2*\L/8,0) {$y$};
42  \draw [fill=red,red] (-\L/2,-\L/2) circle (1pt);
43 \end{tikzpicture}

```

#### 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  $\text{\LaTeX}$ . We refer to our github repository for the source.

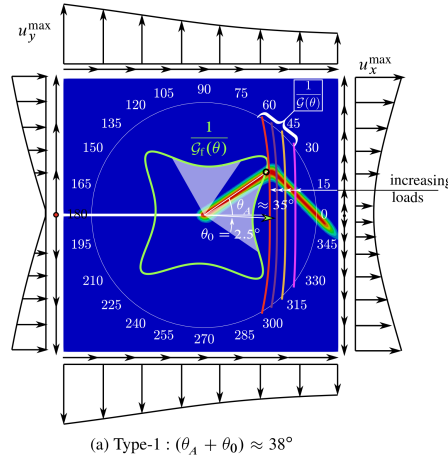


Figure 12: Complex figures: contour plot from ParaView is combined with a polar plot from matplotlib using Inkscape.

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

```

1 \begin{table}[h!]
2 \centering
3 \caption{Material parameters and characteristics for all simulations.}
4 \setlength{\fboxsep}{0pt}
5 \vskip-\topsep%
6 \smallskip%
7 \renewcommand\arraystretch{1.4}
8 \colorbox{darkgray}{%
9 \begin{tabularx}{0.7\textwidth}{111}
10 \toprule
11 Parameter & & Section 5.1 & & Section 5.2 & & \\
12 \midrule
13 Young's modulus [MPa] & & $210 \times 10^3$ & & 145 & & \\
14 Poisson's ratio [-] & & 0.3 & & 0.45 & & \\
15 Tensile strength [MPa] & & 2445 & & 20 & & \\
16 \midrule
17 Experimentally validated & & n/a & & n/a & & \\
18 Solver & & multi-step AM & & single-step AM implicit-explicit & & \\
19 State & & Plane strain & & Plane strain & & \\
20 \bottomrule
21 \end{tabularx}%
22 }
23 \label{table: params}
24 \end{table}

```

### Algorithm 1 Stress update algorithm.

```

1: Inputs:  $\varepsilon_p^t$  (equivalent plastic strain),  $\sigma_t^d$  (un-rotated deviatoric stress),  $\mathbf{d}^d$ , damage  $D^t$ 
2: Outputs:  $\varepsilon_p^{t+\Delta t}$  (equivalent plastic strain),  $\sigma_{t+\Delta t}^d$ 
3: Compute  $G' = (1 - D^t)G$ 
4:  $\sigma_{\text{trial}}^d = \sigma_t^d + 2G'\Delta t \mathbf{d}^d$  ▷ purely elastic stress deviator update
5:  $\sigma_{\text{trial}}^{\text{eq}} = \sqrt{\frac{3}{2}\sigma_{\text{trial}}^d : \sigma_{\text{trial}}^d}$  ▷ equivalent von Mises trial stress
6:  $\sigma_f = \left[ A + B (\varepsilon_p^t)^n \right] \left[ 1 + C \ln \dot{\varepsilon}_p^* \right] \left[ 1 - (T^*)^m \right] (1 - D^t)$  ▷ JC flow stress
7: if  $\sigma_{\text{trial}}^{\text{eq}} < \sigma_f$  then ▷ yielding did not occur, purely elastic step
8:    $\sigma_{n+1}^d = \sigma_{\text{trial}}^d$  26 ▷ keep trial deviatoric stress
9: else ▷ yielding has occurred
10:    $\Delta \varepsilon_p = (\sigma_{\text{trial}}^{\text{eq}} - \sigma_f) / (3G')$  ▷ compute the equivalent plastic strain increment
11:    $\varepsilon_p^{t+\Delta t} = \varepsilon_p^t + \Delta \varepsilon_p$  ▷ update the undamaged matrix plastic strain
12:    $\sigma_{t+\Delta t}^d = \frac{\sigma_f}{\sigma_{\text{trial}}^{\text{eq}}} \sigma_{\text{trial}}^d$  ▷ scale deviatoric stress back to yield surface
13: end if

```

Listing 7: L<sup>A</sup>T<sub>E</sub>X commands to generate Equation (4.1). Each equation should have a unique label.

```

1 \begin{equation}
2   \mathscr{E}(\mathbf{f}, \mathbf{d})
3   = \int_{\varOmega} \left[ \boldsymbol{\omega}(\mathbf{d}) \boldsymbol{\psi}_0^{(\boldsymbol{\epsilon}(\mathbf{f}))} +
4     \boldsymbol{\psi}_0^{(\boldsymbol{\epsilon}(\mathbf{f}))} \right] \mathbf{V} +
5     \int_{\varOmega} \frac{G(\mathbf{f})}{c} \boldsymbol{\alpha} \left[ \frac{1}{b} \boldsymbol{\alpha}(\mathbf{d})
6     + b \left( \boldsymbol{\nabla} \cdot \boldsymbol{\nabla} \right) \right] \mathbf{V}
7     - \mathscr{P}(\mathbf{f})
8 \label{eq:3} % use this label 'eq:3' in \cref{eq:3} to refer to this in text
9 \end{equation}

```

#### 4.6. Source code

Source code can be included in L<sup>A</sup>T<sub>E</sub>X using 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 8 some C++ source code using the Listing package which does not depend on external program. We refer to the L<sup>A</sup>T<sub>E</sub>X source of this paper for the configuration of this package to produce Listing 8.

Listing 8: Presentation of source code using the Listing package with the Bera Mono font (<https://tug.org/FontCatalogue/beramono/>).

```

1 #include <jive/Array.h>
2 #include <jem/base/System.h>
3 using jive::Vector;
4 using jive::Matrix;
5
6 Vector    a;           // double vector Array<double>
7 Matrix    A(10,20);   // double matrix Array<double,2>
8
9 A(slice(BEGIN,3),2) = 2.0; // third col, rows 0, 1, 2 = 2
10 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 9 presents L<sup>A</sup>T<sub>E</sub>X snippets for long figures, tables and equations that span the whole width of the paper. That's all L<sup>A</sup>T<sub>E</sub>X 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.

At this stage you know how to use L<sup>A</sup>T<sub>E</sub>X, 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

Listing 9: Writing two-column papers using L<sup>A</sup>T<sub>E</sub>X.

```

1 \usepackage{mathtools, cuted} % 2 column format, long equation
2
3 % big figure span the whole page
4 \begin{figure*}[]h]
5 \end{figure*}
6
7 % big table span the whole page
8 \begin{table*}[]h]
9 \end{table*}
10
11 % using strip for long equations
12 \begin{strip}
13 % put your long equation here
14 \end{strip}

```

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 scientific content 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.

Now you have submitted our paper and gotten the reviews back, and the editor has asked you to revise your paper taking into account the reviewers’ comments. How should you tackle this task? Different people have different ideas but you can start with Noble (2017) if you do not know where to begin.

## 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 suggestions and recommendations 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 L<sup>A</sup>T<sub>E</sub>X 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 (<https://github.com>).

[com/vinhphunguyen/how-to-write-a-paper](http://com/vinhphunguyen/how-to-write-a-paper)).

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 and learning from them. Our favorite author is Ted Belytschko. We have also learned a lot 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 10 to get PDFs or the one in Listing 11 to get PGF files. PGF pictures 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.

Listing 10: Python script for plotting data.

```

1 import matplotlib as mpl
2 import matplotlib.pyplot as plt
3
4 def set_size(width, fraction=1, subplot=[1, 1]):
5     if width == 'elsevier':
6         width_pt = 468.
7     elif width == 'beamer':
8         width_pt = 307.28987
9     else:
10        width_pt = width
11        fig_width_pt = width_pt * fraction # Width of figure
12        inches_per_pt = 1 / 72.27 # Convert from pt to inches
13        golden_ratio = 0.75 # (5**0.5 - 1) / 2
14        fig_width_in = fig_width_pt * inches_per_pt # Figure width in inches
15        fig_height_in = fig_width_in * golden_ratio * (subplot[0] / subplot[1])
16        fig_dim = (fig_width_in, fig_height_in)
17        return fig_dim
18
19 parser = argparse.ArgumentParser(description='Plot the evolution of variables.',
20                                 add_help=False,
21                                 formatter_class=argparse.RawDescriptionHelpFormatter)
22
23
24 parser.add_argument('--quiet', '-q', action='store_const', const=True, default=False,
25                    help='do not show plot.')
26 args = parser.parse_args()
27
28 # Using seaborn's style
29 plt.style.use('seaborn-colorblind')
30
31 nice_fonts = {
32     "text.usetex": True,
33     "font.family": "serif",
34     "axes.labelsize": 10,
35     "font.size": 10,
36     "legend.fontsize": 8,
37     "xtick.labelsize": 8,
38     "ytick.labelsize": 8,
39 }
40
41 mpl.rcParams.update(nice_fonts)
42
43 # Dimer 5: Read Data
44 fnames = ['log.mpm', 'dam-break-Sun.csv', 'water-break-experiment.csv']
45 dat = [pylab.genfromtxt(f, skip_header=1) for f in fnames]
46
47 fig, ax = plt.subplots(1, 1, figsize=set_size('elsevier', fraction=0.6))
48 # fig, (ax1, ax2) = plt.subplots(1, 2, figsize=set_size(width, subplot=[1, 2]))
49
50 for i, f in enumerate(fnames):
51     x = dat[i][:, 3]
52     y = dat[i][:, 4]
53     ax.plot(x, y, color='black', label='MPM', linewidth=1.5, linestyle='dotted')
54
55     xmax = max(xmax, np.max(x))
56     ymax = max(ymax, np.max(y))
57
58 ax.set_xlabel(r'$T$')
59 ax.set_ylabel(r'$L(T)$')
60 ax.set_ylim(1.0, 2.8)
61 ax.set_xlim(0, 1.6)
62 ax.legend(loc=0)
63 plt.tight_layout()
64 plt.savefig('./water-break-plot.pdf', format='pdf', bbox_inches='tight')
65
66 if args.quiet == False:
67     plt.show()

```

Listing 11: Python script for plotting data.

```

1 nice_fonts = {
2     "text.usetex": True,
3     "font.family": "serif",
4     "pgf.texsystem": "pdflatex",
5     "font.family": "serif",
6     "font.serif": [],
7     "font.sans-serif": [],
8     "font.monospace": [],
9     "pgf.preamble": [
10    # put LaTeX preamble declarations here
11    r"\usepackage[utf8x]{inputenc}",
12    r"\usepackage[T1]{fontenc}",
13    # macros defined here will be available in plots, e.g.:
14    r"\newcommand{\vect}[1]{#1}",
15    # You can use dummy implementations, since you LaTeX document
16    # will render these properly, anyway.
17    ],
18    "axes.labelsize": 10,
19    "font.size": 10,
20    "legend.fontsize": 8,
21    "xtick.labelsize": 8,
22    "ytick.labelsize": 8,
23 }
24 plt.savefig('./cold-spray-plots.pdf', bbox_inches='tight')
25 if args.quiet == False:
26     plt.show()

```

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

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

<sup>†</sup> Footnote 1

<sup>‡</sup> Taken from [Mandal et al. \(2019\)](#).

## 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.

## References

- M. Ambati and L. D. Lorenzis. Phase-field modeling of brittle and ductile fracture in shells with isogeometric NURBS-based solid-shell elements. *Computer Methods in Applied Mechanics and Engineering*, 312:351 – 373, 2016. [Cited on page 10]
- M. Ashby. How to write a paper. *Engineering Department, University of Cambridge, Version*, 5:38, 2000. [Cited on pages 3, 3, and 14]
- R. A. Day. How to write and publish scientific papers, 1998. [Cited on pages 3, 3, and 14]
- 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 pages 11 and 12]
- J. B. C. Engelen. How to include an SVG image in L<sup>A</sup>T<sub>E</sub>X, 2010. URL <http://mirrors.ibiblio.org/CTAN/info/svg-inkscape/InkscapePDFLaTeX.pdf>. [Cited on page 22]
- 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 15 and 15]
- G. D. Gopen and J. A. Swan. The science of scientific writing. *American Scientist*, 78(6):550–558, 1990. [Cited on pages 6 and 7]
- T. Gray. *Publish and flourish: become a prolific scholar*. Teaching Academy, 2005. [Cited on page 18]
- 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 15 and 15]
- R. Huang, N. Sukumar, and J.-H. Prévost. Modeling quasi-static crack growth with the extended finite element method part ii: Numerical applications. *International Journal of Solids and Structures*, 40(26):7539–7552, 2003. [Cited on page 10]
- R. L. Iles. *Guidebook to better medical writing*. Iles publications, 1997. [Cited on page 3]
- 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 pages 6 and 6]
- 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 11 and 31]
- W. S. Noble. Ten simple rules for writing a response to reviewers. *PLoS computational biology*, 13(10), 2017. [Cited on page 28]
- 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 18]
- T. Oetiker, H. Partl, I. Hyna, and E. Schlegl. The not so short introduction to L<sup>A</sup>T<sub>E</sub>X, 2018. URL <http://tug.ctan.org/info/lshort/english/lshort.pdf>. [Cited on page 24]
- 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*, 67:509–513, 2012. [Cited on pages 9 and 17]
- K. W. Plaxco. The art of writing science. *Protein science: a publication of the Protein Society*, 19(12):2261, 2010. [Cited on pages 3 and 14]
- N. P. Rougier, M. Droettboom, and P. E. Bourne. Ten simple rules for better figures. *PLoS computational biology*, 10(9):e1003833, 2014. [Cited on page 20]
- 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 11]