Latex article template

Thomas HOULLIER

v1.0 – July 21, 2022

Abstract

We provide a LaTeX article template. It is meant for scientific writing in article format.

Keywords – LaTeX, Documentation, Article, Technical writing.

Revision History

Revision	Date	Author(s)	Description
1.0	July 21, 2022	TH	creation

Contents

Acronyms

GCD Greatest Common Divisor
JPG Joint Photographic Experts Group
PDF Portable Document Format
PNG Portable Network Graphics
RBF Radial Basis Function
SVG Scalable Vector Graphics

1 Examples

We provide examples of some of the LaTeX features and packages we use most. This is both a visual check that our template formats things the way we want to, and a reference on how to use relevant LaTeX features.

1.1 Citation

We use `BibLaTeX` [3] for bibliography management . The bibliography will automatically only include the references which were cited in the document.

Citations look like:

- Single citation: [9].
- Two citations: [11, 18].
- Citations list: [11, 12, 18]

1.2 Acronyms

We use `glossaries` [10] for acronyms list and eventual glossaries (not included in the present template). Acronyms are defined in a list at the beginning of the document. Only actually used acronyms are included. The acronyms are defined in the text automatically the first time they are used.

For instance:

- First use: Radial Basis Function (RBF).
- Subsequent uses: RBF.

1.3 Equations

Equations are written in the usual manner (see Eqs. (1) to (3)).

$$\int_a^b \omega(x) f(x) \approx \sum_{i=1}^{n} w_i f(x_i) \tag{1}$$

Figure 1: Example vector image in svg format. This is the Inkscape™ logo. Inkscape is a SVG editor [18].

$$\begin{cases} p_{n+1}(x) = (A_n x + B_n) p_n(x) - C_n p_{n-1}(x) \\ p_0(x) = 1 \\ p_1(x) = A_0 x + B_0 \end{cases} \tag{2}$$

$$e^{-at}\cos(\Omega t)u(t) \Leftrightarrow \frac{s+a}{(s+a)^2+\Omega^2} \tag{3}$$

See the LaTeX equation examples at [1, 7].

1.4 Figures

Figures have the following appearance (see Figs. 1 to 3). We use the package `svg` [20] for importing Scalable Vector Graphics (SVG) files.

My own rules of thumb regarding the format of images are:

- **Vector images**, such as SVG: I use SVG as much as possible. I find this important with respect to the feeling of quality a document inspires. I make an exception for images with a number of elements that would be excessive for document rendering.
  - Diagrams: can be edited with Inkscape [18].
  - Plots: can be generated by gnuplot [11].
  - Logos: Most logos have a vector version.
- **Raster images**

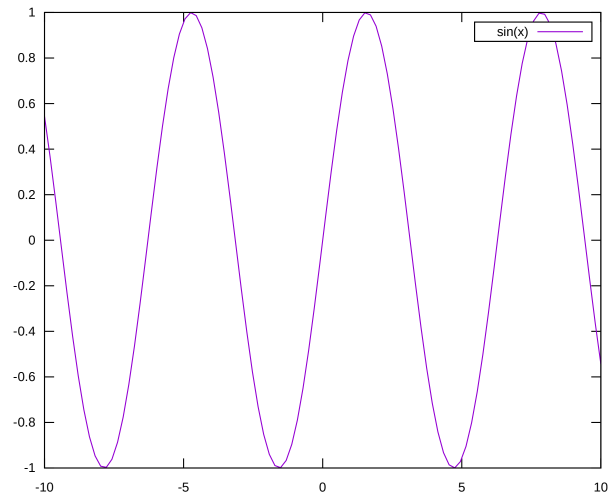

Figure 2: Example plot in svg format. Created using gnuplot [11].

Figure 3: Example raster image: photograph from [12].

1	a	b	c
2	d	e	f
3	g	h	i

Table 1: Example table.

4	j	k	l
5	m	n	o
6	p	q	r

Table 2: Example table: second.

- Photographs: Raster representations are well suited to photographs. Most photographs should be compressed, depending on what the author wants to show, so as not to add unnecessary weight to the output document. The tool jpegoptim [13] is well suited for this purpose. I find both Joint Photographic Experts Group (JPG) and WEBP formats to give good results.
- Screenshots: Screenshots generally contain large areas filled with a single color. Portable Network Graphics (PNG) and WEBP are recommended here.

1.5 Tables

Tables may be created in the usual way (see Tabs. 1 to 3).

1.6 Algorithms

Algorithms use the `algorithm2e` [2] package. Algorithms may be typeset as follows (see Algs. 1 to 3).

1.7 Inline code and Blocks of code

The package `listings` [14] is used to display inline code, verbatim characters and sentences, and blocks of code. It features automatic syntax highlighting for most common programming languages. Commands can be inlined: `cat /etc/group`. Blocks of code are written as Listings 1 to 3.

7	s	t	u
8	v	w	x
9	y	z	χ

Table 3: Example table: third.

Algorithm 1: Euclid's algorithm for Greatest Common Divisor (GCD) [23].

Input : $(A, B) \in \mathbb{N}$
Result: $\gcd(A, B)$

```
X ← A
Y ← B
while X ≠ Y do
    if X > Y then
        X ← X − Y
    else
        Y ← Y − X
return X
```

Algorithm 2: Eratosthenes' Sieve [24].

Input : $n \in \mathbb{N}, n > 1$
Result: Set of prime numbers from 2 up to n

```
A ← array of boolean elements set to True, indexed from 2 up to n.
for i ∈ ⟦2, ⌊√(n)⌋⟧ do
    if A[i] then
        for j ∈ ⟦0, ⌊(n−i²)/i⌋⟧ do
            A[i] ← False
return Every i such that A[i]
```

Algorithm 3: Disjoint Decomposition (sample in [2]).

Input : A bitmap Im of size $w \times l$
Output: A partition of the bitmap

```
special treatment of the first line
for i ← 2 to l do
    special treatment of the first element of line i
    for j ← 2 to w do
        left ← FindCompress(Im[i, j − 1])
        up ← FindCompress(Im[i − 1, ])
        this ← FindCompress(Im[i, j])
        if left compatible with this then // O(left,this)==1
            if left < this then Union(left,this)
            else Union(this,left)
        if up compatible with this then                // O(up,this)==1
            if up < this then Union(up,this)
            // this is put under up to keep tree as flat as
               possible
            else Union(this,up)
            // this linked to up
    foreach element e of the line i do FindCompress(p)
```

```python
>>> signal = np.array([1, 2, 3, 4, 3, 2])
>>> np.fft.fft(signal)
array([15.+0.j,  -4.+0.j,   0.+0.j,  -1.-0.j,   0.+0.j,  -4.+0.j])
    # may vary

>>> np.fft.hfft(signal[:4]) # Input first half of signal
array([15.,  -4.,   0.,  -1.,   0.,  -4.])

>>> np.fft.hfft(signal, 6)  # Input entire signal and truncate
array([15.,  -4.,   0.,  -1.,   0.,  -4.])
```

Listing 1: Python code sample [15].

```
m = rows (A); n = columns (A);
[U, S, V] = svd (A);
## determine the regularization factor alpha
## alpha =
## transform to orthogonal basis
b = U'*b;
## Use the standard formula, replacing A with S.
## S is diagonal, so the following will be very fast and accurate.
x = (S'*S + alpha^2 * eye (n)) \ (S' * b);
## transform to solution basis
x = V*x;
```

Listing 2: Octave code sample [16].

```
(defun entropy (string &aux (length (length string)))
  (declare (type string string))
  (let ((table (make-hash-table)))
    (loop for char across string
          do (incf (gethash char table 0)))
    (- (loop for freq being each hash-value in table
             for freq/length = (/ freq length)
             sum (* freq/length (log freq/length 2))))))
```

Listing 3: Common Lisp code sample [5].

1.8 Footnotes

Footnotes may be included with the native feature[1].

1.9 Epigraphs

> Respondeo dicendum quod necesse est dicere spiritum sanctum a filio esse. Si enim non esset ab eo, nullo modo posset ab eo personaliter distingui.
>
> Sanctus Thomas Aquinas – Summa Theologiae I, q.36, a.2

Epigraphs can be included thanks to `epigraph` [6]. They are meant for the beginning of sections in this template.

1.10 Icons

The `fontawesome` package [8] may be used for small icons in-line with the text .

1.11 Cross-references

We use `cleveref` [4] for cross-referencing. We list cases of cross-referencing to check how they look.

- Equations
  - Single: (Eq. 1).
  - Two: Eqs. (1) and (2).
  - List: Eqs. (1) to (3).
- Figures
  - Single: Fig. 1.
  - Two: Figs. 1 and 2.

[1] Hello

  - List: Figs. 1 to 3

- Tables
  - Single: Tab. 1.
  - Two: Tabs. 1 and 2.
  - List: Tabs. 1 to 3.

- Algorithms
  - Single: Alg. 1
  - Two: Algs. 1 and 2
  - List: Algs. 1 to 3

- Listings
  - Single: Listing 1
  - Two: Listings 1 and 2
  - List: Listings 1 to 3

1.12 Revision history

The revision history at the beginning of this document is created using `vhistory` [22].

2 Usage

Make sure you have a functioning LaTeX installation. We use `TeX Live` [21]. Our target is Portable Document Format (PDF) output.

A `latexmkrc` file provides the boilerplate code to produce the required output, by simply running `latexmk -pdf main.tex`.

3 See also

Further LaTeX template ideas and examples may be found in the following references.

- The Overleaf documentation [17] has many high-quality pages about general LaTeX knowledge.
- Scientific journals provide their LaTeX template for submissions, see for instance Optica [19]. Inspiration may be drawn from these sources.

References

[1] *A collection of LaTeX equations?* URL: https://tex.stackexchange.com/q/119272/215752 Visited on 17 July 2022.

[2] *algorithm2e*. URL: https://ctan.org/pkg/algorithm2e Visited on 20 July 2022.

[3] *BibLaTeX*. URL: https://ctan.org/pkg/biblatex?lang=en Visited on 21 July 2022.

[4] *cleveref*. URL: https://www.ctan.org/pkg/cleveref Visited on 21 July 2022.

[5] *Entropy - Rosetta Code*. URL: https://rosettacode.org/wiki/Entropy#Common_Lisp Visited on 20 July 2022.

[6] *epigraph*. URL: https://ctan.org/pkg/epigraph?lang=en Visited on 21 July 2022.

[7] *Equation sheet*. URL: https://www.equationsheet.com/ Visited on 17 July 2022.

[8] *fontawesome*. URL: https://ctan.org/pkg/fontawesome?lang=en Visited on 17 July 2022.

[9] Neil Gehrels et al. "GALAXY STRATEGY FOR LIGO-VIRGO GRAVITATIONAL WAVE COUNTERPART SEARCHES". In: *The Astrophysical Journal* 820.2 (March 2016), p. 136. DOI: 10.3847/0004-637x/820/2/136.

[10] *glossaries*. URL: https://ctan.org/pkg/glossaries?lang=en Visited on 21 July 2022.

[11] *Gnuplot homepage*. URL: http://www.gnuplot.info/ Visited on 17 July 2022.

[12] Thomas Houllier. *CAD modeling of an alarm clock using FreeCAD*. July 2022. URL: https://github.com/thomashoullier/alarm_clock.

[13] Timo Kokkonen. *jpegoptim*. URL: https://github.com/tjko/jpegoptim.

[14] *listings*. URL: https://ctan.org/pkg/listings?lang=en Visited on 21 July 2022.

[15] *numpy.fft.hfft*. URL: https://numpy.org/doc/stable/reference/generated/numpy.fft.hfft.html Visited on 18 July 2022.

[16] *Octave – Example Code*. URL: https://docs.octave.org/v7.1.0/Example-Code.html Visited on 18 July 2022.

[17] *Overleaf documentation*. URL: https://www.overleaf.com/learn Visited on 21 July 2022.

[18] Inkscape Project. *Inkscape website*. URL: https://inkscape.org/ Visited on 17 July 2022.

[19] *Style Guides and Templates for Optica Publishing Group Journals*. URL: https://opg.optica.org/submit/templates/default.cfm Visited on 21 July 2022.

[20] *svg*. URL: https://ctan.org/pkg/svg?lang=en Visited on 21 July 2022.

[21] *TeX Live*. URL: https://www.tug.org/texlive/ Visited on 21 July 2022.

[22] *vhistory*. URL: https://ctan.org/pkg/vhistory?lang=en Visited on 21 July 2022.

[23] Wikipedia contributors. *Euclidean algorithm — Wikipedia, The Free Encyclopedia*. [Online; accessed 17-July-2022]. 2022. URL: https://en.wikipedia.org/w/index.php?title=Euclidean_algorithm&oldid=1098314618.

[24] Wikipedia contributors. *Sieve of Eratosthenes — Wikipedia, The Free Encyclopedia*. [Online; accessed 20-July-2022]. 2022. URL: https://en.wikipedia.org/w/index.php?title=Sieve_of_Eratosthenes&oldid=1093510438.