

# AstroImages.jl

William Thompson<sup>1</sup> and Mosè Giordano<sup>2</sup>

<sup>1</sup>University of Victoria

<sup>2</sup>Advanced Research Computing, University College London

## ABSTRACT

This is a guide for authors who are preparing papers for JuliaCon using the  $\LaTeX$  document preparation system and the `juliacon` class file.

## Keywords

Julia, Astronomy, Images, World Coordinates

To study the cosmos, astronomers use data cubes with many dimensions representing images with axes for sky position, time, wavelength, polarization, and more. Since these large datasets often span many orders of magnitude in intensity and typically include colours invisible to humans, astronomers like to visualize their images using a variety of non-linear stretching and contrast adjustments. Additionally, images may contain metadata specifying arbitrary mappings of pixel positions to multiple celestial coordinate systems. Julia[1] is a powerful language for processing astronomical data, but these visualization tasks are a challenge for any tool.

*Background.* One of the most ubiquitous data formats used in astronomy is FITS, or the Flexible Image Transport System. Compared to a traditional raster image formats, FITS files are more like containers or small filesystems. Each file contains one or more header data units (HDUs) that pair a dataset with a header. The FITS format and header conventions have developed organically over several decades, first by convention, and then in a series of papers proposing new extensions and standardizing existing behaviour now summarized in [2].

In each HDU, headers are stored in a plain text ASCII format and are followed by data. The data may be in the format of an N-dimensional binary array (an “image” HDU), a binary table, or an ASCII table (“table HDUs”).

The metadata described by FITS headers are quite rich. A header consists of `KEY = VALUE / COMMENT` entries, long form `COMMENT` sections, and long form `HISTORY` sections used to describe the sequence of transformations used to generate the data. FITS headers often contain several hundred header entries most of which are specific to the instrument and software packages used to record and process the data. A subset of these are standardized and used to describe the physical units and coordinates of pixels in the image, called world coordinate system (WCS) header entries.

These WCS entries record pixel locations in one or more celestial coordinate systems including, right ascension and declination, galactic coordinates, velocity, frequency, wavelength, polarization, and more. For most coordinate schemes, pixel coordinates are encoded using an affine transformation matrix combined with a `CTYPE` specifier. Images are typically sampled regularly in the plane

of a detector; however, coordinate projections are in general non-linear, and the physical coordinates of each pixel, spacings between pixels, etc. may vary across the image. In many instances, image HDUs contain 3, 4, or even 5 dimensional data cubes, and thanks to the affine transformation matrix, moving along any one of these dimensions can shift the coordinates along the other dimensions. This point is important since it means that the full coordinates of a given pixel are needed to calculate its physical position along any one axis.

For image data itself, FITS files do not in general store colour information that can be displayed directly. In contrast to digital cameras, astronomical data is almost invariably captured outside the human visible range or at least with filters that are not well matched to human colour perception. Instead, astronomers use visualizations that maps raw numerical data to a false colour image. These steps include a linear transformation from arbitrary numerical ranges to a standard 0 to 255 display range that produces a desired contrast and brightness at an intensity level of interest. The comparatively low dynamic range of electronic displays is often compensated for by applying non-linear mapping, or “stretching” such as log scales or arcsinh scales. Once the data is normalized to a reasonable intensity level, one typically presents a grayscale image or applies a false colour map. Finally, one may also create a colour composite image combining multiple information layers, especially when preparing data for public consumption.

## 1. Introduction

The  $\LaTeX$  document preparation system is a special version of the  $\TeX$ ; typesetting program where-in a collection of  $\TeX$ ; commands are added to  $\LaTeX$  to simplify typesetting. Importantly, it allows the author to concentrate on the logical structure of the document rather than its visual layout.

Moreover,  $\LaTeX$  provides a consistent and comprehensive document preparation interface. There are simple-to-use commands for generating a table of contents, lists of figures and/or tables, and indexes.  $\LaTeX$  can automatically number list entries, equations, figures, tables, and footnotes, as well as articles, sections, and subsections. Using this numbering system, bibliographic citations, page references, and cross references to any other numbered entity (e.g. article, section, equation, figure, list entry, etc.) become quite simple and straightforward. The use of  $\LaTeX$  document classes allows a simple change of class to transform the appearance of your document.

$\LaTeX$  is a powerful tool for managing long and complex documents. In particular, partial processing enables long documents to be produced article by article without losing sequential informa-

tion. The use of document classes allows a simple change of style (or style option) to transform the appearance of your document.

## 2. The JuliaCon Article Class

The `juliacon` class file preserves the standard LATEX interface such that any document that can be produced using the standard LATEX article class can also be produced with the class file.

It is likely that the make up will change after file submission. For this reason, we ask you to ignore details such as slightly long lines, page stretching, or figures falling out of synchronization, as these details can be dealt with at a later stage.

Use should be made of symbolic references (`\ref`) in order to protect against late changes of order, etc.

## 3. USING THE JuliaCon Article CLASS FILE

If the file `juliacon.cls` is not already in the appropriate system directory for L<sup>A</sup>T<sub>E</sub>X files, either arrange for it to be put there or copy it to your working directory. The `juliacon` document class is implemented as a complete class, not a document style option. In order to use the `juliacon` document class, replace `\verbatim` by `juliacon` in the `\documentclass` command at the beginning of your document:

```
\documentclass{article}
```

replace by

```
\documentclass{juliacon}
```

In general, the following standard document style options should *not* be used with the *article* class file:

- (1) `10pt`, `11pt`, `12pt` ? unavailable;
- (2) `twoside` (no associated style file) ? `twoside` is the default;
- (3) `fleqn`, `leqno`, `titlepage` ? should not be used;

## 4. Additional Document Style Options

The following additional style option is available with the `juliacon` class file:

Please place any additional command definitions at the very start of the L<sup>A</sup>T<sub>E</sub>X file, before the `\begin{document}`. For example, user-defined `\def` and `\newcommand` commands that define macros for technical expressions should be placed here. Other author-defined macros should be kept to a minimum.

Commands that differ from the standard L<sup>A</sup>T<sub>E</sub>X interface, or that are provided in addition to the standard interface, are explained in this guide. This guide is not a substitute for the L<sup>A</sup>T<sub>E</sub>X manual itself. Authors planning to submit their papers in L<sup>A</sup>T<sub>E</sub>X are advised to use `juliacon.cls` as early as possible in the creation of their files.

## 5. Additional features

In addition to all the standard L<sup>A</sup>T<sub>E</sub>X design elements, the `juliacon` class file includes the following features: In general, once you have used the additional `juliacon.cls` facilities in your document, do not process it with a standard L<sup>A</sup>T<sub>E</sub>X class file.

### 5.1 Titles, Author's Name, and Affiliation

The title of the article, author's name, and affiliation are used at the beginning of the article (for the main title). These can be produced using the following code:

```
\title{ This is an example of article title }
\author{
  \large 1st Author \[-3pt]
  \normalsize 1st author's affiliation \[-3pt]
  \normalsize 1st line of address \[-3pt]
  \normalsize 2nd line of address \[-3pt]
  \normalsize 1st author's email address \[-3pt]
}
\and
\large 2nd Author \[-3pt]
\normalsize 2nd author's affiliation \[-3pt]
\normalsize 1st line of address \[-3pt]
\normalsize 2nd line of address \[-3pt]
\normalsize 2nd author's email address \[-3pt]
}
\and
\large 3rd Author \[-3pt]
\normalsize 3rd author's affiliation \[-3pt]
\normalsize 1st line of address \[-3pt]
\normalsize 2nd line of address \[-3pt]
\normalsize 3rd author's email address \[-3pt]
}
\maketitle
```

### 5.2 Writing Julia code

A special environment is already defined for Julia code, built on top of *listings* and *jlcode*.

```
\begin{lstlisting}[
  language = Julia,
  numbers=left,
  label={lst:exmplg},
  caption={Example Code Block.}
]
using Plots

x = -3.0:0.01:3.0
y = rand(length(x))
plot(x, y)
\end{lstlisting}
```

Code 1: Example Code Block.

```
1 using Plots
2
3 x = -3.0:0.01:3.0
4 y = rand(length(x))
5 plot(x, y)
```

### 5.3 Abstracts, Key words, term etc...

At the beginning of your article, the title should be generated in the usual way using the `\maketitle` command. For general term and keywords use `\terms`, `\keywords` commands respectively. The abstract should be enclosed within an abstract environment. All these environment can be produced using the following code:

```
\terms{Experimentation, Human Factors}
```

Table 1. : If necessary, the tables can be extended both columns.

Label	Description	Number of Users	Number of Queries
Test 1	Training Data	70	104
Test 2	Testing Data I		105
Test 3	Testing Data II	30	119
	Total	100	328

This is an example of table footnote.

```
\keywords{Face animation, image-based modelling...}
```

```
\begin{abstract}
```

In this paper, we propose a new method for the systematic determination of the model's base of time varying delay system. This method based on the construction of the classification data related to the considered system. The number, the orders, the time delay and the parameters of the local models are generated automatically without any knowledge about the full operating range of the process. The parametric identification of the local models is realized by a new recursive algorithm for on line identification of systems with unknown time delay. The proposed algorithm allows simultaneous estimation of time delay and parameters of discrete-time systems. The effectiveness of the new method has been illustrated through simulation.

```
\end{abstract}
```

## 6. Some guidelines

The following notes may help you achieve the best effects with the `juliacon` class file.

### 6.1 Sections

$\text{\LaTeX} 2_{\epsilon}$  provides four levels of section headings and they are all defined in the `juliacon` class file:

```
—\section
—\subsection
—\subsubsection
—\paragraph
```

Section headings are automatically converted to allcaps style.

### 6.2 Lists

The `juliacon` class file provides unnumbered lists using the `unnumlist` environment for example,

First unnumbered item which has no label and is indented from the left margin.

Second unnumbered item.

Third unnumbered item.

The unnumbered list which has no label and is indented from the left margin. was produced by:

```
\begin{unnumlist}
\item First unnumbered item...
\item Second unnumbered item...
\item Third unnumbered item...
\end{unnumlist}
```

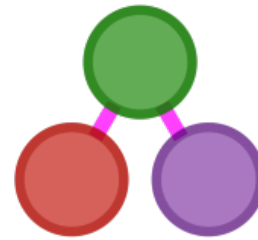


Fig. 1: This is example of the image in a column.

The `juliacon` class file also provides hyphen list using the `itemize` environment for example,

- First unnumbered bulleted item which has no label and is indented from the left margin.
- Second unnumbered bulleted item.
- Third unnumbered bulleted item which has no label and is indented from the left margin.

was produced by:

```
\begin{itemize}
\item First item...
\item Second item...
\item Third item...
\end{itemize}
```

Numbered list is also provided in `acmtog` class file using the `enumerate` environment for example,

- (1) The attenuated and diluted stellar radiation.
- (2) Scattered radiation, and
- (3) Reradiation from other grains.

was produced by:

```
\begin{enumerate}
\item The attenuated...
\item Scattered radiation, and...
\item Reradiation from other grains...
\end{enumerate}
```

### 6.3 Illustrations (or figures)

The `juliacon` class file will cope with most of the positioning of your illustrations and you should not normally use the optional positional qualifiers on the `figure` environment that would override these decisions.

The figure 1 is taken from the JuliaGraphs organization <sup>1</sup>.

<sup>1</sup><https://github.com/JuliaGraphs>

Table 2. : Tuning Set and Testing Set

Label	Description	Number of Users	Number of Queries
Test 1	Training Data	70	104
Test 2	Testing Data I		105
Test 3	Testing Data II	30	119
	Total	100	328

Figure captions should be *below* the figure itself, therefore the `\caption` command should appear after the figure or space left for an illustration. For example, Figure 1 is produced using the following commands:

```
\begin{figure}
\centerline{\includegraphics[width=20pc]{Graphics.eps}}
\caption{An example of the testing process for a
binary tree. The global hypothesis is tested
first at level  $\alpha$  (a), and the level of
individual variables is reached last (d). Note
that individual hypotheses can be tested at
level  $\alpha/4$  and not  $\alpha/8$  as one might
expect at first.}
\label{sample-figure_2}
\end{figure}
```

Figures can be resized using first and second argument of `\includegraphics` command. First argument is used for modifying figure height and the second argument is used for modifying figure width respectively.

Cross-referencing of figures, tables, and numbered, displayed equations using the `\label` and `\ref` commands is encouraged. For example, in referencing Figure 1 above, we used `Figure~\ref{sample-figure}`

## 6.4 Tables

The `juliacon` class file will cope with most of the positioning of your tables and you should not normally use the optional positional qualifiers on the table environment which would override these decisions. Table captions should be at the top.

```
\begin{table}
\tbl{Tuning Set and Testing Set}{
\begin{tabular}{|l|l|c|c|}\hline
Label & \multicolumn{1}{c}{Description}
& Number of Users &
Number of Queries\\\hline
Train70 & Training Data &
Test70 & Testing Data I & 105\\\hline
Test30 & Testing Data II & 30 & 119\\\hline
& Total & 100 & 328\\\hline
\end{tabular}}
\end{table}
```

## 6.5 Landscaping Pages

If a table is too wide to fit the standard measure, it may be turned, with its caption, to 90 degrees. Landscape tables cannot be produced directly using the `juliacon` class file because  $\text{\LaTeX}$  itself cannot turn the page, and not all device drivers provide such a facility. The following procedure can be used to produce such pages.

Use the package `rotating` in your document and change the coding from

```
\begin{table}...\end{table}
to
\begin{sidewaystable}...\end{sidewaystable}
and for figures
\begin{figure}...\end{figure}
to
\begin{sidewaysfigure}...\end{sidewaysfigure}
```

environments in your document to turn your table on the appropriate page of your document. For instance, the following code prints a page with the running head, a message half way down and the table number towards the bottom.

```
\begin{sidewaystable}
\tbl[Landscape table caption to go here.]{...}
\label{landtab}
\end{sidewaystable}
```

## 6.6 Double Column Figure and Tables

For generating the output of figures and tables in double column we can use the following coding:

(1) For Figures:

```
\begin{figure*}...\end{figure*}
```

(2) For landscape figures:

```
\begin{sidewaysfigure*}...\end{sidewaysfigure*}
```

(3) For Tables:

```
\begin{table*}...\end{table*}
```

(4) For landscape tables:

```
\begin{sidewaystable*}...\end{sidewaystable*}
```

## 6.7 Typesetting Mathematics

The `juliacon` class file will set displayed mathematics with center to the column width, provided that you use the  $\text{\LaTeX}$  standard of open and closed square brackets as delimiters. The equation

$$\sum_{i=1}^p \lambda_i = (S)$$

was typeset using the `acmtog` class file with the commands

```
\[
\sum_{i=1}^p \lambda_i = (S)
\]
```

For display equations, cross-referencing is encouraged. For example,

```
\begin{equation}
(n-1)^{-1} \sum_{i=1}^n (X_i - \overline{X})^2.
\label{eq:samplevar}
\end{equation}
```

`Equation~(\ref{eq:samplevar})` gives the formula for sample variance.

The following output is generated with the above coding:

$$(n-1)^{-1} \sum_{i=1}^n (X_i - \overline{X})^2. \quad (1)$$

Equation (1) gives the formula for sample variance.

## 6.8 Enunciations

The `juliacon` class file generates the enunciations with the help of the following commands:

```
\begin{theorem}...\end{theorem}
\begin{strategy}...\end{strategy}
\begin{property}...\end{property}
\begin{proposition}...\end{proposition}
\begin{lemma}...\end{lemma}
\begin{example}...\end{example}
\begin{proof}...\end{proof}
\begin{definition}...\end{definition}
\begin{algorithm}...\end{algorithm}
\begin{remark}...\end{remark}
```

The above-mentioned coding can also include optional arguments such as

```
\begin{theorem}[...]. Example for theorem:
\begin{theorem}[Generalized Poincare Conjecture]
Four score and seven ... created equal.
\end{theorem}
```

THEOREM 1 GENERALIZED POINCARÉ CONJECTURE. *Four score and seven years ago our fathers brought forth, upon this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.*

## 6.9 Extract

Extract environment should be coded within

```
\begin{extract}...\end{extract}
```

## 6.10 Balancing column at last page

For balancing the both column length at last page use :

```
\vadjust{\vfill\pagebreak}
```

at appropriate place in your  $\text{\TeX}$  file or in bibliography file.

## 7. Handling references

References are most easily (and correctly) generated using the BIB-TEX, which is easily invoked via

```
\bibliographystyle{juliacon}
\bibliography{ref}
```

When submitting the document source (`.tex`) file to external parties, the `ref.bib` file should be sent with it. [1]

## 8. References

- [1] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017. doi:10.1137/141000671.
- [2] FITS Working Group. Definition of the flexible image transport system (FITS). Technical report, International Astronomical Union, 2016.