Archive of Numerical Software
vol. 1, no. S1, pages 1–6
© 2011

Preparing articles for submission to the Archive of Numerical Software

Editor O. Archive*[1] and Author O. Software[2]

[1]The Archive of Numerical Software
[2]The Worlds Best Place for Numerical Software

Received: Jan 1, 1677; **final revision:** Jun 1, 1677; **published:** Jun 13, 1677.

Abstract: This document describes the general structure of the `ansarticle` document class and serves as an example for its usage. The `ansarticle` document class is based on the standard LaTeX `article` document class and a few other standard packages. It fixes style parameters to the preferred style of the Archive of Numerical Software (ANS).

1 Requirements

To write an article for ANS, you need to download the following files from the ANS web page:

- `paper.tex`, the source file for this document and a template for authors;
- `ansarticle.cls`, the ANS document class file;
- `anslistings.sty`, the ANS style file for typesetting of code;
- `bibliography.bib`, a bibliography with some important references;
- `Makefile`, useful for building your document.

These files are all included in the tarball `ansarticle.tgz`.

The file `ansarticle.cls` loads the standard LaTeX `article.cls` and some additional packages. It also sets up the layout. Hardly any of the options of the packages are available to the author, in order to keep the journal style consistent.

Upon loading, `ansarticle.cls` requires the following packages: `a4wide`, `algorithm`, `algorithmicx`, `algpseudocode`, `authblk`, `amsmath`, `amssymb`, `booktabs`, `fancyhdr`, `graphicx`, `hyperref`, `lastpage`, `listings`, `todonotes`. These packages are part of standard LaTeX distributions and should not cause any difficulties.

*Additional thanks to my neighbor's dog for waking me up on time to work on this style file

Entity	Dimension	Codimension
Vertex	0	D
Edge	1	$D-1$
Face	2	$D-2$
Facet	$D-1$	1
Cell	D	0

Table 1: Mesh entities and their dimensions/codimensions.

2 Usage

2.1 Building the document

Use this file (`paper.tex`) as a template for your article. Then simply type `make` to build your article. This will call `pdflatex` to generate the PDF file `paper.pdf`. To also call `bibtex`, type `make final`.

2.2 Typesetting mathematical formulæ

The ANS document class uses the `amsmath` and `amssymb` packages for improved typesetting of mathematical formulælike, e.g., this one:

$$\frac{1}{\sqrt{2}+\frac{1}{\sqrt{2}+\frac{1}{\sqrt{2}+\cdots}}}. \tag{1}$$

Make sure that you familiarize yourself with the `amsmath` environments `multline`, `align` and `split` (and know the differences between them). Avoid home-made splitting using `{array}{rcl}`. For details, refer to the `amsmath` documentation.

2.3 Typesetting tables

Use the commands provided by the `booktabs` package for better looking typesetting of tables, as illustrated in Table 1.

2.4 Typesetting algorithms

Use the functionality provided by the packages `algorithm`, `algorithmic` and `algpseudocode`, automatically included by the `ansarticle` document class, to typeset algorithms. This is illustrated in Algorithm 1.

2.5 Typesetting code

All code listings are based on the `listings` package [?]. The package `anslistings`, automatically included by `ansarticle` document class, provides standard styles for a variety of common programming languages. Changing any style parameters for this package is not encouraged and may lead to unexpected outcomes in the final publication.

Below, we provide samples of listings for the some of the languages available. If a language is missing, please contact the managing editor for an updated style file.

Algorithm 1 Compute all triangular orthogonal polynomials up to degree d by recurrence

```
$D^{0,0}(x,y) := 1$
$D^{1,0}(x,y) := \frac{1+2x+y}{2}$
for $p \leftarrow 1, d-1$ do
    $D^{p+1,0}(x,y) := \left(\frac{2p+1}{p+1}\right)\left(\frac{1+2x+y}{2}\right) D^{p,0}(x,y) - \left(\frac{p}{p+1}\right)\left(\frac{1-y}{2}\right)^2 D^{p-1,0}(x,y)$
end for
for $p \leftarrow 0, d-1$ do
    $D^{p,1}(x,y) := D^{p,0}(x,y)\left(\frac{1+2p+(3+2p)y}{2}\right)$
end for
for $p \leftarrow 0, d-1$ do
    for $q \leftarrow 1, d-p-1$ do
        $D^{p,q+1}(x,y) := \left(a_q^{2p+1,0} y + b_q^{2p+1,0}\right) D^{p,q}(x,y) - c_q^{2p+1,0} D^{p,q-1}(x,y)$
    end for
end for
```

2.5.1 C++ Code should be compliant with the current C++ standard. Two commands are provided to prettyprint C++ code. First, an environment to put C++ code into the LaTeX file, namely

```
\begin{c++}
  // Your C++ code here
\end{c++}
```

Alternatively, it is possible to use `\inputcpp{file.cc}` to print the C++ code in a file. Here is an example of typesetting of C++ code:

C++ code

```cpp
// Get dimensions of local mesh_data
const unsigned int num_local_cells = mesh_data.cell_vertices.size();
assert(global_cell_indices.size() == num_local_cells);
const unsigned int num_cell_vertices = mesh_data.cell_vertices[0].size();

// Build array of cell-vertex connectivity and partition vector
std::vector<unsigned int> cell_vertices;
std::vector<unsigned int> cell_vertices_partition;
const unsigned int size = num_local_cells*(num_cell_vertices + 1);
cell_vertices.reserve(size);
cell_vertices_partition.reserve(size);
for (unsigned int i = 0; i < num_local_cells; i++)
{
  cell_vertices.push_back(global_cell_indices[i]);
  cell_vertices_partition.push_back(cell_partition[i]);
  for (unsigned int j = 0; j < num_cell_vertices; j++)
  {
    cell_vertices.push_back(mesh_data.cell_vertices[i][j]);
    cell_vertices_partition.push_back(cell_partition[i]);
  }
}

// Distribute cell-vertex connectivity
MPI::distribute(cell_vertices, cell_vertices_partition);
assert(cell_vertices.size());
cell_vertices_partition.clear();
```

2.5.2 Python The following illustrates typesetting of Python code:

Python code

```python
# Time-stepping
t = dt
while t < T:

    # Compute tentative velocity step
    b1 = assemble(L1)
    [bc.apply(A1, b1) for bc in bcu]
    solve(A1, u1.vector(), b1, "gmres", "ilu")

    # Pressure correction
    b2 = assemble(L2)
    [bc.apply(A2, b2) for bc in bcp]
    solve(A2, p1.vector(), b2, "gmres", "amg_hypre")

    # Velocity correction
    b3 = assemble(L3)
    [bc.apply(A3, b3) for bc in bcu]
    solve(A3, u1.vector(), b3, "gmres", "ilu")

    # Plot solution
    plot(p1, title="Pressure", rescale=True)
    plot(u1, title="Velocity", rescale=True)

    # Save to file
    ufile << u1
    pfile << p1

    # Move to next time step
    u0.assign(u1)
    t += dt
```

2.6 Handling supplementary material

Additional material which is not part of the article text, but part of the submission, for instance setup scripts or video output, can be linked from the article, using the `hyperref` commands (`hyperref` is automaticly included by `ansarticle`). URLs to such material should start with the filename. A suitable prefix will be added to the links in the editorial process.

2.7 Citations

The `ansarticle` document class imports the `natbib` package, which provides the commands `\citet{}` and `\citep{}`. These should be used in place of the regular `\cite{}` command as illustrated by the following passage: It was found by **?** that what we cannot speak about we must pass over in silence, contrary to common belief [**?**].

2.8 Other

The `ansarticle` document class provides a number of useful notation macros which you can find by browsing through the file `ansarticle.cls`.

2.9 Checklist

Before submitting your manuscript to ANS, please review the following checklist:

1. Have you run a spell checker on your manuscript (using an American English dictionary)?
2. Have you referenced all libraries you have been using and the appropriate versions and descriptive publications?
3. Have you referred to all figures and tables (callouts)?

4. Have you used a scalable vector graphics format for all images where possible?

Before submitting your source code to ANS, please review the following checklist:

1. Does a simple 'make' or similar build and run the program in a default configuration?
2. Did you include all configuration information necessary?
3. Did you include instructions on how to reproduce the examples in the text?

3 Production after article was accepted

3.1 The manuscript

Once the article is accepted, the editor will send you a file `ansinfo.tex` in the same directory. The file contains macros defining the publication information like in the following listing:

```
\def\ANSreceived{Jan 1, 1677}
\def\ANSrevised{Jun 1, 1677}
\def\ANSaccepted{Jun 12, 1677}
\def\ANSpublished{Jun 13, 1677}
\def\ANSvolume{1}
\def\ANSnumber{S1}
\def\ANSyear{2011}
```

Please copy this file where LaTeX can find it and recompile your manuscript to produce the finale version of the PDF.

3.2 The source code

If necessary, clean up the source code directory. Make sure you include only the files readers should download with the source (see the next subsection on media files). Make sure, a simple 'make' or similar will run a first example. Assert that the data in the manuscript can be reproduced.

3.3 The media directory

Every article comes with a media directory. This directory contains a file `index.html` which is linked from the article page and should serve as an entry point to your media files. The minimum requirement on this file is the following head:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD␣XHTML␣1.0␣Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>
  <title>ANS V(N) Authors: Title</title>
  <link rel="stylesheet" type="text/css" href="/ans.css"/>
</head>

<body>
  <h2>Archive of Numerical Software V(N)</h2>
  <h1>Title</h1>
  <h2>Authors</h2>

  Text and links
</body>
</html>
```

The occurences of `V(N)` should be replaced by the assigned volume and number of the publication, `Title` and `Authors` accordingly. You are welcome to use image tags in this file.

Add your image files and movies to this directory, not to the source code!

4 Philosophiæ Naturalis Principia Mathematica

Also add main references for all libraries listed on the ANS web page.

As an example text, we present an excerpt from **?**.

4.1 Definitiones

4.1.1 Def. I Quantitas Materiæ est mensura ejusdem orta ex illius Densitate & Magnitudine conjunctim.

4.1.1.1 Aer duplo densior in duplo spatio quadruplus est. Idem intellige de Nive et Pulveribus per compressionem vel liquefactionem condensatis. Et par est ratio corporum omnium, quæ per causas quascunq; diversimode condensantur. Medii interea, si quod fuerit, interstitia partium libere pervadentis, hic nullam rationem habeo. Hanc autem quantitatem sub nomine corporis vel Massæ in sequentibus passim intelligo. Innotescit ea per corporis cujusq; pondus. Nam ponderi proportionalem esse reperi per experimenta pendulorum accuratissime instituta, uti posthac docebitur.

4.1.2 Def. II Quantitas motus est mensura ejusdem orta ex Velocitate et quantitate Materiæ conjunctim.

4.1.2.1 Motus totius est summa motuum in partibus singulis, adeoq; in corpore duplo majore æquali cum Velocitate duplus est, et dupla cum Velocitate quadruplus.

4.1.3 Def. III Materiæ vis insita est potentia resistendi, qua corpus unumquodq;, quantum in se est, perseverat in statu suo vel quiescendi vel movendi uniformiter in directum.

4.1.3.1 Hæc semper proportionalis est suo corpori, neq; differt quicquam ab inertia Massæ, nisi in modo concipiendi. Per inertiam materiæ fit ut corpus omne de statu suo vel quiescendi vel movendi difficulter deturbetur. Unde etiam vis insita nomine significantissimo vis inertiæ dici possit. Exercet vero corpus hanc vim solummodo in mutatione status sui per vim aliam in se impressam facta, estq; exercitium ejus sub diverso respectu et Resistentia et Impetus: Resistentia quatenus corpus ad conservandum statum suum reluctatur vi impressæ; Impetus quatenus corpus idem, vi resistentis obstaculi difficulter cedendo, conatur statum ejus mutare. Vulgus Resistentiam quiescentibus et Impetum moventibus tribuit; sed motus et quies, uti vulgo concipiuntur, respectu solo distinguuntur ab invicem, neq; semper vere quiescunt quæ vulgo tanquam quiescentia spectantur.

4.1.3.1.1 A subparagraph If you really want to nest that deeply, go ahead! But, while you might have enjoyed reading **??**, you are encouraged to question the applicability to a publication on a thing as profane as software.