

# Guidelines for Preparing a Paper for AISB 2008

Name1 Surname1 and Name2 Surname2 and Name3 Surname3<sup>1</sup>

**Abstract.** The purpose of this paper is to show a contributor the required style for a paper for AISB 2008. The specifications for layout are described so that non- $\text{\LaTeX}$  users can create their own style sheet to achieve the same layout. The source for the sample file is available for  $\text{\LaTeX}$  users. The PostScript and the PDF file is available for all.

## 1 PAGE LIMIT

The page limit for AISB 2008 papers depends on the individual symposium (see symposium Web page).

## 2 GENERAL SPECIFICATIONS

The following details should allow contributors to set up the general page description for their paper:

1. The paper is set in two columns each 20.5 picas (86 mm) wide with a column separator of 1.5 picas (6 mm).
2. The typeface is Times Modern Roman.
3. The body text size is 9 point (3.15 mm) on a body of 11 point (3.85 mm) (i.e. 61 lines of text).
4. The effective text height for each page is 56 picas (237 mm). The first page has less text height. It requires an additional footer space of 3.5 picas (14.8 mm) for the copyright inserted by the publisher and 1.5 picas (6 mm) of space before the title. The effective text height of the first page is 51 picas (216 mm).
5. There are no running feet for the final camera-ready version of the paper. The submission paper should have page numbers in the running feet.

## 3 TITLE, AUTHOR, AFFILIATION, COPYRIGHT AND RUNNING FEET

### 3.1 Title

The title is set in 20 point (7 mm) bold with leading of 22 point (7.7 mm), centered over the full text measure, with 1.5 picas (6 mm) of space before and after.

### 3.2 Author

The author's name is set in 11 point (3.85 mm) bold with leading of 12 point (4.2 mm), centered over full text measure, with 1.5 picas (6 mm) of space below. A footnote indicator is set in 11 point (3.85 mm) medium and positioned as a superscript character.

---

<sup>1</sup> University of Leipzig, Germany, email: somename@informatik.uni-leipzig.de

### 3.3 Affiliation

The affiliation is set as a footnote to the first column. This is set in 8 point (2.8 mm) medium with leading of 8.6 point (3.1 mm), with a 1 point (0.35 mm) footnote rule to column width.

### 3.4 Copyright

The copyright details will be inserted by the publisher.

### 3.5 Running feet

The running feet are inserted by the publisher. For submission you may insert page numbers in the middle of the running feet. Do not, however, insert page numbers for the camera-ready version of the paper.

## 4 ABSTRACT

The abstract for the paper is set in 9 point (3.15 mm) medium, on a body of 10 point (3.5 mm). The word Abstract is set in bold, followed by a full point and a 0.5 pica space.

## 5 HEADINGS

Three heading levels have been specified:

### 1. A level headings

- The first level of heading is set in 11 point (3.85 mm) bold, on a body of 12 point (4.2 mm), 1.5 lines of space above and 0.5 lines of space below.
- The heading is numbered to one digit with a 1 pica space separating it from the text.
- The text is keyed in capitals and is unjustified.

### 2. B level headings

- The second level of heading is set in 11 point (3.85 mm) bold, on a body of 12 point (4.2 mm), 1.5 lines of space above and 0.5 lines of space below.
- The heading is numbered to two digits separated with a full point, with a 1 pica space separating it from the text.
- The text is keyed in upper and lower case with an initial capital for first word only, and is unjustified.

### 3. C level headings

- The third level of heading is set in 10 point (3.5 mm) italic, on a body of 11 point (3.85 mm), 1.5 lines of space above and 0.5 lines of space below.

- The heading is numbered to three digits separated with a full point, with a 1 pica space separating it from the text.
- The text is keyed in upper and lower case with an initial capital for first word only, and is unjustified.

#### 4. Acknowledgements

## 12 THEOREMS

The text of a theorem is set in 9 point (3.15 mm) italic on a leading of 11 point (3.85 mm). The word Theorem and its number are set in 9 point (3.15 mm) bold.

A one line space separates the theorem from surrounding text.

**Theorem 1** *Let us assume this is a valid theorem. In reality it is a piece of text set in the theorem environment.*

## 13 FOOTNOTES

Footnotes are set in 8 point (2.8 mm) medium with leading of 8.6 point (3.1 mm), with a 1 point (0.35 mm) footnote rule to column width<sup>2</sup>.

## 14 REFERENCES

The reference identifier in the text is set as a sequential number in square brackets. The reference entry itself is set in 8 point (2.8 mm) with a leading of 10 point (3.5 mm), and appears in the sequence in which it is cited in the paper.

## 15 SAMPLE CODING

The remainder of this paper contains examples of the specifications detailed above and can be used for reference if required.

## 16 PROGRAMMING MODEL

Our algorithms were implemented using the *single program, multiple data* model (SPMD). SPMD involves writing a single code that will run on all the processors co-operating on a task. The data are partitioned among the processors which know what portions of the data they will work on [7].

### 16.1 Structure of processes and processors

The grid has  $P = P_r \times P_c$  processors, where  $P_r$  is the number of rows of processors and  $P_c$  is the number of columns of processors.

#### 16.1.1 Routing information on the grid

A message may be either *broadcast* or *specific*. A broadcast message originates on a processor and is relayed through the network until it reaches all other processors. A specific message is one that is directed to a particular target processor.

Broadcast messages originate from a processor called *central* which is situated in the ‘middle’ of the grid. This processor has coordinates  $(\lfloor P_r/2 \rfloor, \lfloor P_c/2 \rfloor)$ . Messages are broadcast using the *row-column broadcast* algorithm (RCB), which uses the following strategy. The number of steps required to complete the RCB algorithm (i.e. until all processors have received the broadcast value) is given by  $\lfloor P_r/2 \rfloor + \lfloor P_c/2 \rfloor$ .

A specific message is routed through the processors using the *find-row-find-column* algorithm (FRFC) detailed in [5]. The message is sent from the *originator* processor vertically until it reaches a processor sitting in the same row as the *target* processor. The message is then moved horizontally across the processors in that row until it

reaches the target processor. An accumulation based on the recursive doubling technique [9, pp. 56–61], would require the same number of steps as the RCB requires. If either the row or column of the originator and target processors are the same then the message will travel only in a horizontal or vertical direction, respectively (see [12]).

## 17 DATA PARTITIONING

We use *data partitioning by contiguity*, defined in the following way. To partition the data (i.e. vectors and matrices) among the processors, we divide the set of variables  $V = \{i\}_{i=1}^N$  into  $P$  subsets  $\{W_p\}_{p=1}^P$  of  $s = N/P$  elements each. We assume without loss of generality that  $N$  is an integer multiple of  $P$ . We define each subset as  $W_p = \{(p-1)s + j\}_{j=1}^s$  (see [11], [4] and [2] for details).

Each processor  $p$  is responsible for performing the computations over the variables contained in  $W_p$ . In the case of vector operations, each processor will hold segments of  $s$  variables. The data partitioning for operations involving matrices is discussed in Section 18.3.

## 18 LINEAR ALGEBRA OPERATIONS

### 18.1 Saxpy

The saxpy  $w = u + \alpha v$  operation, where  $u$ ,  $v$  and  $w$  are vectors and  $\alpha$  is a scalar value, has the characteristic that its computation is *disjoint elementwise* with respect to  $u$ ,  $v$  and  $w$ . This means that we can compute a saxpy without any communication between processors; the resulting vector  $w$  does not need to be distributed among the processors. Parallelism is exploited in the saxpy by the fact that  $P$  processors will compute the same operation with a substantially smaller amount of data. The saxpy is computed as

$$w_i = u_i + \alpha v_i, \quad \forall i \in \{W_p\}_{p=1}^P \quad (4)$$

### 18.2 Inner-product and vector 2-norm

The inner-product  $\alpha = u^T v = \sum_{i=1}^N u_i v_i$  is an operation that involves accumulation of data, implying a high level of communication between all processors. The mesh topology and the processes architecture used allowed a more efficient use of the processors than, for instance, a ring topology, reducing the time that processors are idle waiting for the computed inner-product value to arrive, but the problem still remains. The use of the SPMD paradigm also implies the global broadcast of the final computed value to all processors.

The inner-product is computed in three distinct phases. Phase 1 is the computation of partial sums of the form

$$\alpha_p = \sum_{\forall i \in \{W_p\}} u_i \times v_i, \quad p = 1, \dots, P \quad (5)$$

The accumulation phase of the inner-product using the RCA algorithm is completed in the same number of steps as the RCB algorithm (Section 16.1.1). This is because of the need to relay partial values between processors without any accumulation taking place, owing to the connectivity of the grid topology.

The vector 2-norm  $\alpha = \|u\|_2 = \sqrt{u^T u}$  is computed using the inner-product algorithm described above. Once the inner-product value is received by a processor during the final broadcast phase, it computes the square root of that value giving the required 2-norm value.

<sup>2</sup> This is an example of a footnote that occurs in the text. If the text runs to two lines the second line aligns with the start of text in the first line.

### 18.3 Matrix–vector product

For the matrix–vector product  $v = Au$ , we use a *column partitioning* of  $A$ . Each processor holds a set  $W_p$  (see Section 17) of  $s$  columns each of  $N$  elements of  $A$  and  $s$  elements of  $u$ . The  $s$  elements of  $u$  stored locally have a one-to-one correspondence to the  $s$  columns of  $A$  (e.g. a processor holding element  $u_j$  also holds the  $j$ -th column of  $A$ ). Note that whereas we have  $A$  partitioned by columns among the processors, the matrix–vector product is to be computed by *rows*.

The algorithm for computing the matrix–vector product using column partitioning is a generalization of the inner-product algorithm