

The digiconfigs package

Lars Madsen^{*,†}

June 22, 2006

1 Introduction

In Stochastic Geometry and Digital Image Analysis (both fields in statistics) some problems can be solved in terms of so-called *configurations*. Which is basically a square matrix with \circ and \bullet in them (and interpreted in a specific fashion, that I do not know about).

The configurations are mainly used to estimate geometric features, such as the mean normal measure, the area density, and the length density, of digitized random sets in the plane. Other applications include the restoration of noisy realisations of random sets.
— pseudo-anonymous researcher

Now, a certain type of problem might need more than 50×5 matrices of \circ 's and \bullet 's, which is not very handy to write in the traditional way.

```
\begin{bmatrix}
\circ & \bullet & \circ & \circ & \bullet \\
\dots \\
\end{bmatrix}
```

Of course one method of easing this (for 5×5 matrices) is to create a macro with five arguments, which would then create the matrix. (Something like `\macro{o . . oo}{ooo . o}{. . . . o}{.}{o . o . o}`.) But that does not scale very well, we need a macro that given a special string (and perhaps a matrix size configuration) can simply build a configuration matrix.

The digiconfigs-package tries to solve this problem.

2 Usage

Simply load the package in the traditional way:

```
\usepackage{digiconfigs}
```

This will also load the amsmath package. Digiconfigs provides the `\dconfig` macro, which has the following syntax:

```
\dconfig[<matrix-size>]{<configuration string>}
```

*Email: daleif@imf.au.dk

†Version 0.5

Explanation:

<matrix-size> This optional argument is used to specify the expected size of the matrix, the syntax to be used is of course $\langle rows \rangle \times \langle cols \rangle$, where, if $\langle rows \rangle = \langle cols \rangle$ is equivalent to simply typing $\langle rows \rangle$. Default is 4 (i.e. a 4×4 matrix), which is configurable.

<configuration string> This is suppose to be a string of single characters each of which will represent an entry in the matrix. The macro will react to four different characters, which by default is

- o** represents a \circ ,
- .** represents a \bullet ,
- c** represents a \cdot (sometimes used to indicate a blank), and
- b** which represents a *blank* entry.

All four characters are configurable, as is the symbols used for *circle*, *bullet*, and *cdot*.

All other characters (except #, % and X) will be ignored. X is internally added to the string, and is used to detect the end of the string if it falls short.

`\dconfig` will then calculate the number of entries it has to look for, and simply eat up character by character (using tail recursion) until it has found the number it needs. Any remaining characters will be ignored. If the string is too short for the expected number of entries, a warning will be issued, and the matrix will simply come short at this point.

Here are a few examples

```
\def\mystring{.o.. bb.o.. dsfsf co..}
\[
\dconfig[2]{\mystring} \dconfig[1x5]{\mystring}
\dconfig{\mystring} \dconfig[3x4]{\mystring}
\]
```

$$\begin{pmatrix} \bullet & \bullet \\ \circ & \bullet \end{pmatrix} (\bullet \quad \bullet \quad \circ \quad \bullet \quad \bullet) \begin{pmatrix} \bullet & \bullet & \circ & \bullet \\ \bullet & & & \bullet \\ \circ & \bullet & \bullet & \cdot \\ \circ & \bullet & \bullet & \end{pmatrix} \begin{pmatrix} \bullet & \bullet & \circ & \bullet \\ \bullet & \bullet & \bullet & \cdot \end{pmatrix}$$

Matrix three here falls short of entries.

3 Configuration

The package offers various configurations. All of these can be changed throughout the document.

```
\setCircleChar{<char>} (o)
\setBulletChar{<char>} (.)
\setCdotChar{<char>} (c)
\setBlankChar{<char>} (b)
```

Is used to alter the four main characters the macro is looking for. Argument should be a single char (to keep things simple).

```

\setCircleSymbol{<symbol>} (\circ)
\setBulletSymbol{<symbol>} (\bullet)
\setCdotSymbol{<symbol>} (\cdot)

```

Is used to alter the output symbols used in the matrix. The *blank* symbol cannot be changed.

```

\setMatrixStart{<matrix start code>} (\begin{bmatrix})
\setMatrixEnd{<matrix end code>} (\end{bmatrix})

```

Fairly obvious. So if you want to save some space use

```

\setMatrixStart{\left[\begin{smallmatrix}]
\setMatrixEnd{\end{smallmatrix}\right]}

```

```

\setDefaultMatrixSize{<matrix size>} (4)

```

Alters the expected number of entries. Use a positive integer or the $\langle number \rangle \times \langle number \rangle$ -syntax.

4 Example

```

\setMatrixStart{\left[\begin{smallmatrix}]
\setMatrixEnd{\end{smallmatrix}\right]}
\setCircleChar{+}
\setBulletChar{-}
\setCircleSymbol{1}
\setBulletSymbol{-1}
\[
\dconfig{c-+c +-+- -+-- b-+b -+--}
\]

```

$$\begin{bmatrix} \cdot & -1 & 1 & \cdot \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & -1 \end{bmatrix}$$

5 A tip

Here is a small tip. If you find `\dconfig` to be a bit long, then try something like

```

\newcommand\dc[2][\dconfig[#1]{#2}]

```